DEPICTER: Deep representation clustering for histology annotation
Eduard Chelebian a,∗, Christophe Avenel a, Francesco Ciompi b, Carolina Wählby a

Keywords:
Interactive annotation
Histology
Self-supervised learning
Clustering

A B S T R A C T

Automatic segmentation of histopathology whole-slide images (WSI) usually involves supervised training of deep learning models with pixel-level labels to classify each pixel of the WSI into tissue regions such as benign or cancerous. However, fully supervised segmentation requires large-scale data manually annotated by experts, which can be expensive and time-consuming to obtain. Non-fully supervised methods, ranging from semi-supervised to unsupervised, have been proposed to address this issue and have been successful in WSI segmentation tasks. But these methods have mainly been focused on technical advancements in algorithmic performance rather than on the development of practical tools that could be used by pathologists or researchers in real-world scenarios. In contrast, we present DEPICTER (Deep rEPresentatIon ClusTERing), an interactive segmentation tool for histopathology annotation that produces a patch-wise dense segmentation map at WSI level. The interactive nature of DEPICTER leverages self- and semi-supervised learning approaches to allow the user to participate in the segmentation producing reliable results while reducing the workload.

DEPICTER consists of three steps: first, a pretrained model is used to compute embeddings from image patches. Next, the user selects a number of benign and cancerous patches from the multi-resolution image. Finally, guided by the deep representations, label propagation is achieved using our novel seeded iterative clustering method or by directly interacting with the embedding space via feature space gating. We report both real-time interaction results with three pathologists and evaluate the performance on three public cancer classification dataset benchmarks through simulations. The code and demos of DEPICTER are publicly available at https://github.com/eduardchelebian/depicter.

1. Introduction

1.1. Background and contributions

Recent progress in histopathology whole-slide image (WSI) segmentation was due to advances in fully supervised learning [1–3]. It involves training deep learning models with pixel-level annotated labels for classifying each WSI pixel into distinct tissue types. The U-Net, a convolutional neural network (CNN) with skip connections introduced by Ronneberger et al. [4], has consistently achieved state-of-the-art results in histopathology segmentation. For instance, Oskal et al. [5] used it for automated epidermis segmentation; or Hermsen et al. [6] applied multiple U-Nets for multi-class kidney tissue segmentation. The main challenge of fully supervised segmentation lies in the resource-intensive generation of large, pixel-level annotated datasets for effective model learning and generalization. To address this limitation, two overlapping strategies can be employed: developing methods that do not require pixel-level annotations on every slide or simplifying the annotation process to generate training data.

Alternatives to fully supervised methods have been proposed for overcoming these challenges [7,8]. In the next section, we explore successful approaches, ranging from semi-supervised to unsupervised, for WSI segmentation. These methods achieve results comparable to fully supervised ones by handling scenarios with incomplete supervision, where annotation are not available throughout the entire WSI dataset. Despite their importance, these methods are typically presented as technical breakthroughs rather than practical tools for pathologists or researchers.

Only fast enough methods suitable for seamless on-the-fly annotation can be incorporated into interactive tools [9–12]. However, such tools often have one or more limitations: (1) they sacrifice deep learning power for speed, (2) they require retraining based on user input, challenging for gigapixel-sized WSI on local computers, (3) they focus on nuclei segmentation rather than tissue detection or (4) they are proof-of-concept tools not available to researchers.

To meet the need for an open-source, computationally efficient interactive tool utilizing deep learning for WSI tissue annotation, we
introduce DEPICTER (Deep rEPrEsentatIon ClusTERing). Our tool incorporates self-supervised and semi-supervised approaches into an interactive annotation tool, generating patch-wise dense segmentation at the WSI-level from partial annotations. It can function as a rapid patch-level annotation generator to speed up the production of training material or as a standalone tissue annotation tool, aiding researchers or pathologists in compartmentalizing WSI into different morphological entities.

DEPICTER consists of three steps: first, a CNN encoder computes meaningful image representations of WSI using either a pretrained or contrastive self-supervised network. Second, we show an overlay of patch coordinates on top of the full-resolution WSI along with a dimensionality-reduced feature space, allowing users to select tissue regions. This study validates DEPICTER on the clinically relevant task of benign against tumor segmentation. Finally, initial labels are propagated by a novel semi-supervised seeded iterative clustering or by manual feature gating in the reduced space. DEPICTER is implemented as a plugin for the open-source visualization software TissU-Maps 3.01 to leverage its browser-based and GPU-accelerated visualization capabilities.

The following is a summary of our contributions:

• We introduce DEPICTER, an interactive and web-based tool to annotate histopathology images. The user selects a few patches and, either by semi-supervised propagation or manual feature gating of the feature space, it results in a patch-wise dense segmentation of the morphologically relevant entities based on their deep representations.

• We propose seeded iterative clustering, a semi-supervised clustering algorithm for label propagation based on positive seed reclustering.

• We provide user experience metrics from trained histopathologists such as time-to-accuracy and amount of interactions, to give an idea of the annotation effort when using DEPICTER.

• We evaluate on three large-scale cancer segmentation benchmarks for breast sentinel lymph node, lung and digestive system histopathology slides, achieving competitive results and showing how DEPICTER can be used broadly across tissues.

• We compare pretraining methods for a clustering task, which allows us to directly discern the specific effects of the representation space they produce, not biased by further downstream training.

DEPICTER is an extension of our clustering algorithm presented at a NeurIPS 2022 workshop [14]. In this study, we embedded a fast implementation of the algorithm, adding interactivity, post-processing and optimizing pretraining methods and architectures. Furthermore, we extensively evaluated DEPICTER on three public histopathology cancer detection benchmarks (CAMELYON17, ACDC and DigestPath) and ran usability tests with three pathologists.

1.2. Related work

Segmentation can range in the requirement of annotation effort and user interaction. In this section, we categorize related work accordingly.

1.2.1. Non-fully supervised segmentation

Many approaches have been proposed as an alternative for the fully supervised framework for image segmentation. In this section, we review the ones that have been used for histopathology image segmentation without full supervision. For a complete and general view, we refer the reader to other reviews on the topic by Cheplygina et al. [7] or Peng and Wang [15].

Semi-supervised. These approaches require less labeled data by combining abundant unlabeled data, easier to acquire, with smaller amounts of labeled data. In segmentation scenarios, it means using pixel-wise annotations on a limited subset of the dataset. In this direction, Li et al. [16] trained an expectation maximization-based semi-supervised approach to segment prostate cancer WSI using 135 fully annotated images and 1800 with only slide-level annotations. Similarly, Xie et al. [17] used a pairwise relation-based semi-supervised approach for gland segmentation on WSI, achieving results close to fully-supervised models with 50% of the dataset fully annotated.

Although these results showed the potential of leveraging unlabeled data, they rely on dense pixel annotations, demanding intensive manual effort and limiting an interactive tool.

Partially- and inaccurately-supervised. The partially-supervised scenario, a more challenging subset of semi-supervised methods, involves only partial or sparse annotations such as scribbles, points or clicks on the different class regions of the image. Note that they are correct but not dense throughout the whole image. Qiu et al. [18] applied this approach to segment nuclei using a combination of semi-supervised and self-training strategies. In Bokhorst et al. [19], a modified loss function enabled training CNN with sparse annotations, achieving comparable performance to full supervision in colorectal cancer segmentation.

Inaccurately-supervised methods address coarse and noisy labels, presenting the most challenging scenario while also allowing the user to save more time. Han et al. [20] employed patch-level pseudo masks to yield pixel-level maps for lung adenocarcinoma, acknowledging that a single patch can contain pixels from different classes. Guo et al. [21] proposed SAC-Net for nuclei segmentation based on noisy pseudo labels simulated by Voronoi diagrams.

Our proposed method would fall into this category, as it allows to develop interactive tools that leverage partial and patch-level annotations where the user can give feedback but still not need to annotate a complete WSI.

Weakly- and unsupervised. Weakly-supervised approaches refer to situations with only WSI-level weak labels available, with no pixel-level spatial information. Xu et al. [22] proposed CAMEL, using multiple-instance learning and achieving results close to fully supervised approaches on a colorectal adenoma and the CAMELYON16 datasets. Another approach by Zhang et al. [23] used a joint graph and CNN network for both H&E and immunohistochemistry WSI.

Unsupervised learning poses the most challenging segmentation scenario as it operates without label information. Khan et al. [24] extracted traditional tissue morphology features and used clustering to segment breast WSI into hypo- and hyper-cellular stroma. More recently, Xu et al. [25] proposed TisCut, which employs object clustering and Voronoi diagrams for segmentation in necrosis and melanoma datasets, achieving up to 80% Jaccard index coefficients.

While these approaches require minimal pathologist input, they lack control during the process. Improvement relies on hyperparameter tuning rather than direct image interaction, potentially leading to suboptimal results.

1.2.2. Self-supervised learning

Self-supervised learning is a pretraining method that has gained popularity in non-fully-supervised scenarios for its ability to learn relevant features using unlabeled data from proxy tasks [8]. Some of the most commonly used approaches in the field are SimCLR [26] and MoCo [27], based on contrastive learning or DINO [28], which uses student–teacher distillation.

Ciga et al. [29] demonstrated the efficacy of SimCLR pretraining on 57 histopathology datasets, outperforming the commonly used ImageNet pretraining. Chen and Krishnan [30] compared SimCLR and DINO in histopathology scenarios, finding that DINO vision transformers also learn relevant and interpretable features. Subsequent work by Chen et al. [31] presented a vision transformer using the intrinsic pyramidal organization in WSI for learning representations. An interesting application by Chen et al. [32] used self-supervised learning to guide histopathology image search and retrieval.

1 https://tissuumaps.github.io.
1.2.3. Interactive segmentation tools

After exploring non-fully supervised frameworks for histology image segmentation, in this section we focus on computational tools for interactively annotating such images.

Many methods are intended for segmenting specific objects, such as cell foreground against background mainly in fluorescence microscopy images for tasks such as cell tracking. Widely used tools include Trainable Weka Segmentation [9], ielastic [11] and LABKIT [12]. While some, like Trainable Weka Segmentation, NuClick [33] or QuickAnnotator [34], extend to histopathology WSI, their application is confined to specific ROI and need active retraining, making them unsuitable for large-scale tasks like tumor classification.

However, tools specifically designed for tumor histopathology WSI exist. TissueWand [35] presented extensive analysis of design choices, but lacks available software. Another approach [36] used deep learning to iteratively guide a support vector machine for segmentation and evaluated it on tumor masks from CAMELYON16 and TCGA data. However, there was no available implementation as well. DeepScribble [37] proposed a two-stage classification and interactive refinement approach but also lacks implementation.

A notable mention is HistoCloud [38], designed for specific tasks using pretrained models. While not intended for segmentation on partial annotations, it offers a user-friendly interface for interactive output correction.

To our knowledge, the only available software for interactive tumor segmentation based on sparse annotations on WSI is QuPath [10]. The well-established software includes a pixel classifier, enabling users to train a model interactively on scribbles for pixel-based segmentation of WSI. However, the classification is based on classical features such as Gaussian filters, limiting its ability to identify cancer.

2. Methods

In the following sections we present each part of DEPICTER: the generation of deep representations, the seeded iterative clustering algorithm and the post-processing step using conditional random fields. Additionally, we present how DEPICTER can be used to annotate slides by manual gating in feature space. Fig. 1 shows the complete workflow.

2.1. Deep embeddings by self-supervised learning

The following section will showcase self-supervised pretrained with ResNet18 as a case study but, as shown in the ablation studies in Section 3.3, other choices could be made.

After dividing the whole-slide images in the dataset into non-overlapping patches, we chose to use contrastive self-supervised learning SimCLR [26] as a pre-training method. SimCLR is based on agreement maximization of the activations $z$ of two augmentations of the same patch by means of the normalized temperature scaled cross entropy loss (NT-Xent) for a pair of examples $(i, j)$:

$$ l_{ij} = -\log \frac{\exp(\text{sim}(z_i, z_j) / \tau)}{\sum_{k=1}^{2N} I_{[k \neq i]} \exp(\text{sim}(z_k, z_j) / \tau)} \ (1) $$

where $\text{sim}(z_i, z_j)$ represents the cosine similarity function between the activated projections of two patches, $\tau$ is a temperature parameter and $I_{[k \neq i]}$ is the indicator function.

Apart from the usual augmentations, we chose to use an H&E specific color augmentation proposed by Otálora et al. [39].2 For each dataset, we fine-tuned a ResNet18 model using the already pretrained model proposed by Ciga et al. [29] as a starting point.

After pretrained the network, we extract the penultimate layer representations before the prediction layer such that, in the end, we will have a set of embedding vectors $H = \{h_1, \ldots, h_N\}$ for $N$ patches, where each $h_i$ will be a vector whose length will depend on the number of activations in the specific architecture (e.g. ResNet18 will produce 512 features after pooling in the last layer, so $H \in \mathbb{R}^{N \times 512}$). For notation purposes, we will refer to the set of embedding vectors as $X$, as they will be the input to our method.

2.1.1. Visualization

Drawing inspiration from the spatial transcriptomics field, we employ the technique of overlaying the coordinates of the patch centers onto the multi-resolution image as distinct spots, using TissU-Umaps [13]. To facilitate an intuitive visualization, we adopt the common approach of displaying the image on one side and the two-dimensional Uniform Manifold Approximation and Projection (UMAP) [40] representations of the extracted features on the other. This visual arrangement is shown in Figs. 2a and b. With this strategy, we enable the user to interact with both spaces simultaneously, selecting the spots within one space and seeing their corresponding coordinates highlighted in the other space. Such integrated scheme will prove valuable for both the seed placing mechanism (Fig. 2c) and the feature space annotation functionality (Fig. 2d), which will be introduced in the following sections.

2.2. Seeded iterative clustering

We propose to extend the idea of seeded k-means by re-clustering the target class in a binary clustering scenario [14]. The high level idea for re-clustering only the positive class comes from the observation that the positive class is usually over-clustered with this kind of methods. For the specific problem of tumor tissue as the positive class, the negative class non-cancerous tissue can have many different ways of manifesting. Thus, re-clustering can be seen as a way of eliminating one entity of the non-cancerous background at a time.

To showcase our method, we will briefly formalize the k-means and the extended seeded k-means algorithms. To this end, we will use the notation in the original seeded k-means paper by Basu [41].

Given a set of points $X = \{x_1, \ldots, x_N\}$, k-means identifies $K$ clusters $\{X_k\}_{k=1}^K$ of $X$, based on $K$ randomly initialized cluster centers $\{\mu_1, \ldots, \mu_K\}$ that are iteratively refined by minimizing the objective function

$$ F = \sum_{l=1}^{K} \sum_{x_i \in X_l} \left\| x_i - \mu_l \right\|^2. \ (2) $$

The extension by Basu [41] introduced the concept of seeding by additionally defining a seed set $S \subseteq X$, where for each partition $X_l$ there is at least one $x_i \in S$. Essentially, the only difference with the original algorithm is that instead of initializing the cluster centers $\{\mu_1, \ldots, \mu_K\}$ randomly, each $\mu_i$ is initialized with the mean of $S_i \in S$.

Although the seeds are not used at any other point, the algorithm becomes semi-supervised. As the authors point out, from a Bayesian perspective, one can consider that the seed set determines the conditional distribution on which the expectation step is calculated.

We propose seeded iterative clustering (SIC) to further refine seeded clustering in a binary problem, such as cancer detection, by iteratively re-clustering the target class i.e. cancerous samples. Given $\mathcal{X} = \{x_1, \ldots, x_N\}$ and a $S \subseteq \mathcal{X}$ where we have at least one $x_i \in S$ for both $X_0$ (benign) and $X_1$ (cancerous): (1) we first initialize the labels $\hat{y}^{(0)}$ by running the seeded k-means algorithm as described before; (2) successively, we perform iterative clustering and label updates until the loss $\mathcal{L}$ no longer improves. As this method does not need to have a differentiable loss for gradient descent but it is rather more similar to a hierarchical decision tree, we can use directly the accuracy score as our loss $\mathcal{L}$, simply:

$$ \mathcal{L} = \frac{1}{|S|} \sum_{i \in S} [y_i = \hat{y}_i]. \ (3) $$

Fig. 1. Complete workflow of DEPICTER. (a) The contrastive self-supervised learning SimCLR workflow pretrains the model by maximizing the agreement between the activations $z_i, z_j$ of two augmented versions $x_i, x_j$ of each patch $x$. (b) This model is then used to extract a set of representations $H = h_1, \ldots, h_N$ for all $N$ patches. (c) The pathologist selects seeds (positive in magenta and negative in blue) $S$ which guide the (d) seeded iterative clustering of representations $H$ (referred to as $K$ as explained in Section 2.1). (e) A step of post-processing using conditional random fields further refines the results. Steps c-e are iterated and guided by a UMAP representation of $H$ as described in Fig. 2.

Table 1

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set of data points $X$ and set of initial seeds $S$</td>
<td>$K$ clusters of $X$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>Initialize number of clusters $K = 2$ and iterations $t = 0$</td>
</tr>
<tr>
<td>2:</td>
<td>Repeat until convergence of $C$:</td>
</tr>
<tr>
<td>2a:</td>
<td>Compute $(\mu_i, \mu_j)^{(t)}$ according to $(S_0, S_1)^{(t)}$, if $S \neq \emptyset$ else randomly and skip 2c</td>
</tr>
<tr>
<td>2b:</td>
<td>Assign each $x_i$ to the closest cluster according to $C$ to obtain two sets ${X_0, X_1}^{(t)}$</td>
</tr>
<tr>
<td>2c:</td>
<td>Update $X_0^{(t+1)} = X_0^{(t)}$ and $S^{(t+1)} = (S_0^{(t)} \subseteq H^{(t)})$</td>
</tr>
<tr>
<td>3:</td>
<td>$t \leftarrow (t + 1)$</td>
</tr>
</tbody>
</table>

where $|S|$ is the size of the sample set $S$, $y_i$ is the true label of the $i$th sample in $S$, and $\hat{y}_i$ is the predicted label for the same sample.

At each iteration $t$, we update the labels based on the results of seeded k-means clustering, applied only to the samples in $X_1$. We repeat until there are no more points in $S_1$, at which point we compute an additional non-seeded k-means on the $X_1$ to test if the score further improves.

Finally, we return the predicted labels $\hat{y}_i$ of the latest best score. The complete pseudo code is presented in Table 1.

2.3. Conditional random fields

Conditional random fields (CRF) are graphical models that have been commonly used in the computer vision community as a post-processing method in image segmentation to add smoothness and context terms to ensure label agreement. Following the notation in Krähenbühl and Koltun [42], the Gibbs energy that characterizes a fully connected CRF is defined by

$$E(x) = \sum_{i} \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j)$$

where $\psi_u(x_i)$ represents the unary potentials obtained from the certainty we attribute to the seeded iterative clustering algorithm labels $y_i$, and

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) k(f_i, f_j)$$

represents the pairwise potentials modeled by the label compatibility function $\mu$ that controls the penalty for pixels which are close in the defined pairwise potentials $k$ but are in different clusters. As for the kernels defined over the features $f_i, f_j$, we used the same smoothness kernel defined in the original publication but, instead of defining our appearance kernel on the RGB colors of the image, we used the 2D dimensionality reduced space of the deep embedding.
gating technique further enhances the versatility of DEPICTER. The integration of this feature space investigation of potential relationships between them and their spatial and identification of different tissue compartments, as well as the logical profiles. This annotation strategy facilitates the exploration to define distinct subsets or compartments based on their morpho-dimensional UMAP representation, as shown in Fig. 2f, we allow users compartments such as cancerous and benign regions.

By manually drawing regions of interest directly on the two-dimensional UMAP representation, as shown in Fig. 2f, we allow users to define distinct subsets or compartments based on their morphological profiles. This annotation strategy facilitates the exploration and identification of different tissue compartments, as well as the investigation of potential relationships between them and their spatial distribution within the sample. The integration of this feature space gating technique further enhances the versatility of DEPICTER.

3. Experiments and results

3.1. Datasets

We tested DEPICTER on three public datasets of different characteristics in terms of size and resolution, to highlight different aspects of the method: CAMELYON17, DigestPath and ACDC.

DEPICTER is intended to ease the annotation burden for large lesions. Using DEPICTER to annotate small lesions, where the lesion size is similar to the total area of the patches required for seeding, would bias the evaluation of the performance of the model. Information regarding lesion size was available in the CAMELYON17 dataset, and slides labeled as small lesions were excluded from the evaluation. Information on lesion size was not available for the ACDC and DigestPath datasets. Instead, we excluded the cases where the lesion area covered ten patches or less, understanding that there will not be a gain in efficiency by the pathologist, as they could annotate the whole image with less than ten clicks. Ten clicks is in the lower range of number of interactions by a pathologist (as observed in Fig. 3a). Examples of excluded slides for all datasets are included in Supplementary Figure 1.

A summary of the datasets is provided in Table 2.

CAMELYON17. Litjens et al. [43]. The Cancer MEtastases in LYmph nOdes challenge 2017 (CAMELYON17) dataset is a collection of high-resolution histopathology images of lymph nodes with annotations for the presence of metastatic breast cancer lesions, immunohistochemically-stained for cytokeratin to confirm the classification. They are annotated as macro-metastases, micro-metastases, or isolated tumor cells. Therefore, out of the 100 publicly available images from the CAMELYON17 dataset we selected the 17 with macro lesions for our experiments.

ACDC. Li et al. [44]. The Automatic Cancer Detection and Classification in whole-slide Lung Histopathology challenge (ACDC@LungHP) dataset is a collection of whole-slide images of lung tissue annotated with ground truth segmentation labels for the cancerous tissue. From the 150 publicly available WSI in the training set, we filtered the 116 cases that contained tumor annotations and, from those, the 104 cases with tumors covering more than ten patches.

DigestPath. Da et al. [45]. The Digestive-System Pathological Detection and Segmentation challenge (DigestPath) colonoscopy tissue segment dataset is a collection of colonoscopy tissue slides annotated for benign and malignant areas. The training set contains 250 images from 93 WSI from which we again selected the cases where the tumor area covered at least ten patches. This resulted in 196 images being finally included in the study.

### Table 2

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Raw resolution</th>
<th>Eligible cases</th>
<th>Evaluated cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAMELYON17</td>
<td>~100,000(^2) pixels</td>
<td>40x</td>
<td>100</td>
<td>17</td>
</tr>
<tr>
<td>ACDC</td>
<td>~100,000(^2) pixels</td>
<td>20x</td>
<td>150</td>
<td>104</td>
</tr>
<tr>
<td>DigestPath</td>
<td>~1000(^2) pixels</td>
<td>20x</td>
<td>250</td>
<td>196</td>
</tr>
</tbody>
</table>

3.1.1. Patch extraction and annotation

Each of the datasets are divided into 224 × 224 non-overlapping patches at the level of the resolution pyramid corresponding to 10×. As the segmentation masks are available at pixel level and we are working at patch level, we need to find a way of translating them. We define a patch as positive if the center pixel falls within the annotated region at the level of the pyramid at which the patches are extracted. The rationale behind using the center pixel as a criterion for labeling is to facilitate the annotation process. When visualizing the patches,
DEPICTER overlays the center coordinates on the image. This design allows pathologists to adjust marker size while maintaining the same center position. By selecting the center pixel to determine the label, we ensure that pathologists can intuitively mark the center, irrespective of any adjustments to marker size, if they identify the patch as positive.

3.2. Evaluation of DEPICTER

3.2.1. Real-time interaction

To test the usability of DEPICTER and whether the connection between the UMAP feature space and the image can be exploited by a pathologist, we designed an experiment to test the feature space gating-based annotations. We asked three pathologists (two highly trained pathologists and one resident pathologist) to use DEPICTER on the 17 slides of the CAMELYON17 dataset. The typical workflow would include:

1. Identifying and selecting a number of patches inside and outside malignant regions in the image.
2. Seeing where the patches are highlighted in the feature space and manually drawing gates in feature space.
3. Further correcting by clicking or drawing in both the image and feature spaces.

This enables us to calculate (1) the time of the first interaction, which gives us a surrogate for the time needed by the pathologist to find the lesion, (2) the time of the last interaction, which gives the total time to use the tool, (3) the number of interactions (such as clicking and gating) and (4) the Dice–Sørensen coefficient ($DSC$) and the binary Cohen’s kappa coefficient ($\kappa$) of the final segmentation result:

\[
DSC = \frac{2TP}{2TP + FP + TN}
\]

\[
\kappa = \frac{2(TP \cdot TN - FP \cdot FN)}{(TP + FP)(FP + TN)(TP + FN)(FN + TN)}
\]

where $TP$, $TN$, $FP$, $FN$ are true positives, true negatives, false positives and false negatives, respectively.

Note that the sessions with the pathologists lasted about one hour and the pathologists were learning to use DEPICTER as they made the annotation, with minimal training involved.

3.2.2. Simulations

To scale up the evaluation to bigger datasets, we designed a regime that simulates a pathologist selecting one positive patch and one negative patch and then improving the results by iterative correction of random patches:

1. Randomly assign two patches as seeds -one in the cancerous region and another in the benign region- and run DEPICTER.
2. Correct one random patch that was incorrectly clustered for each class and run DEPICTER again.
3. Repeat the last step four more times and check whether the results improve.

This means that the results presented are obtained with a maximum of five positive selections and five negative selections, corresponding to ten interactions. We calculate the $DSC$ after ten interactions and provide the cumulative distribution of cases that reached the same results at initialization and after iterative corrections.

3.3. Ablation studies

To test the influence of the different elements of the model, we used the same regime as in the simulation studies and varied the model by:

(i) Using or not the post-processing CRF step for the slides that will benefit from the step.
(iii) Choosing different pre-training methods for the same ResNet18 architecture: using the ImageNet weights directly, using publicly available weights pretrained using SimCLR on 57 histopathology datasets in Ciga et al. [29] (SimCLR public) or using the SimCLR pretraining on each dataset with SimCLR public as a starting point (SimCLR own).

For each experiment we evaluate the evolution of the $DSC$ and Cohen’s $\kappa$.

3.4. Comparison with existing methods

We performed a direct comparison with the annotation functionality of QuPath. From the real-time interaction experiments with the pathologists, we identified two samples from the CAMELYON17 dataset as being the least and most challenging to annotate. Using these, we ran DEPICTER and QuPath’s pixel classification module using the same seeds. As DEPICTER works patch-wise, to compare them it was necessary to transform the patch seeds to pixels and transform the pixel classifier’s output to patches. We would like to clarify that this is not a comparison of TissUUmaps with QuPath, which are the tools hosting the annotation modules; DEPICTER could be equally implemented within QuPath, which is a step we consider as future work but out of scope for the current paper.

Comparing an interactive method like DEPICTER with previously successful fully supervised approaches presents a challenge due to a fundamental difference in their respective training procedures. While fully supervised methods utilize the training set for parameter optimization and report results on the test set, our approach can only evaluate results on the publicly available positive cases, which are found in the training set. Despite this limitation, we believe that the comparison is still valuable as it provides insight into the performance range of our method.

The best performing algorithm according to the CAMELYON17 paper [46] achieved a $\kappa$ score of 0.8993 [47]. The ACDC challenge had two tracks, using either a single model which achieved a $DSC$ of 0.7544 or multi-model approaches, achieving a higher $DSC$ of 0.8372 by combining the DenseNets and dilation block to work with UNets. None had accompanying papers available. Finally, the best result for DigestPath was achieved by Zhu et al. [48] ($DSC = 0.8075$) using an adversarial context-aware and appearance consistency UNet. To assess the performance of our interactive method in comparison to previously successful fully supervised approaches, we report the same metrics used by the studies, namely the $DSC$ for ACDC and DigestPath and Cohen’s $\kappa$ for CAMELYON17.

3.5. Results

Fig. 3 shows the results of the interaction experiments for the feature space gating-based annotation with the three pathologists. In under an hour, including a small training session, the pathologists achieved a Cohen’s $\kappa$ of 0.893S, 0.9088 and 0.9135 on the CAMELYON17 training subset with macro lesions. They took an average of about 2 min per slide, including the time for finding the actual lesion. Considering the limitations of annotating in a patch-basis and evaluating a patch as positive when the center pixel is positive, and also the limited time allocated per slide, the results are very competitive. We can observe as well how similarly difficult cases tend to be grouped closely (i.e. cases 7 and 8) and how some cases seem to have a hard limit irrespective of how many interactions and time it took to analyze (i.e. cases 2 and 16). The average number of interactions, initial and final times and performances can be found in Supplementary Table 1. The actual distribution...
of interactions per slide and pathologist is shown in Supplementary Figure 3.

Table 3 shows the results of the ablation studies. We can see that applying the CRF post-processing in general has positive results, even if the improvement is not very consistent among all the experiments, improving the DSC for CAMELYON17 using ImageNet pretrained ResNet18 from 0.7505 to 0.8605, while only marginally improving from 0.6824 to 0.6931 for DigestPath using the same model.

The choice of architecture between the different depths of ResNet seems to favor the smallest architecture, ResNet18, along almost every experiment. The differences are especially clear on CAMELYON17 and DigestPath, while on ACDC the choice of architecture did not seem to affect the results as much, with a DSC of 0.7012, 0.7081 and 0.6989 for ResNet18, 50 and 101, respectively using the features from the ImageNet pretrained model without postprocessing using CRF.

The most consistent differences are seen when using the same architecture pretrained on ImageNet or using contrastive self-supervised learning via SimCLR. Both the SimCLR public and SimCLR own, pretrained on histology images, outperformed the models trained on natural images. Fig. 4 compares the features of a ACDC image extracted using ResNet18 pretrained on ImageNet and pretrained using SimCLR on the whole ACDC dataset. It is clear that the different morphological entities are better clustered when using self-supervised learning. This is especially apparent in the patches that contain both cancer and benign tissue, which are grouped more sparsely in the ImageNet features.

In every dataset and for every metric SimCLR own pretrained on the dataset at hand performed the best. Across the datasets, the SimCLR own model with CRF performed the best in the CAMELYON17 dataset with DSC = 0.9031 and a κ score showing almost perfect agreement, while in ACDC and DigestPath it performed similarly achieving DSC = 0.7388 and DSC = 0.7433, with a κ score in the moderate agreement bracket. The box-plots with the distribution of results for these features are shown in Supplementary Figure 2.

In Fig. 5 we show how the results improve with the number of corrections. For easier comparison, the y-axis is normalized considering the result after four corrections. We see that more than 50% of the CAMELYON17 cases achieved the same result as with four corrections, that is two positives and two negative seeds, while DigestPath and ACDC needed one extra correction for a big jump in the percentage of cases reaching the same result as with four corrections.

The results from the comparison with QuPath’s pixel classifier can be found in Supplementary Figure 4. The pixel classifier is not designed to handle classification at high resolutions locally, which is why we applied it to lower resolution images. Even though it does identify part of the tumor, there is a substantial amount of false positives caused by the method not capturing subtle enough morphological differences. Conversely, DEPICTER allows defining the negative class more broadly, independent of the actual entity, as it has been designed to recluster the positive class iteratively, thus helping with reducing the false positives.

Finally, it is not straightforward how to compare these results to the best results in every challenge, as the metrics are provided pixel-wise and on the test set while we provide them patch-wise and on a subset of the training set. With this in mind, in Table 4 we show comparable

---

Table 3 Average DSC and Cohen’s kappa values for the ablation study for different comparisons (CRF: Conditional Random fields).

<table>
<thead>
<tr>
<th>Methods</th>
<th>CAMELYON17</th>
<th>ACDC</th>
<th>DigestPath</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSC Kappa</td>
<td>DSC Kappa</td>
<td>DSC Kappa</td>
</tr>
<tr>
<td><strong>ImageNet w/o CRF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.7505 0.7404</td>
<td>0.7012 0.4632</td>
<td>0.6824 0.4860</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.6954 0.6733</td>
<td>0.7081 0.4732</td>
<td>0.6620 0.4571</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.7359 0.7171</td>
<td>0.6989 0.4561</td>
<td>0.6659 0.4609</td>
</tr>
<tr>
<td><strong>SimCLR w/o CRF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet18 public</td>
<td>0.8457 0.8366</td>
<td>0.7157 0.4906</td>
<td>0.7120 0.5310</td>
</tr>
<tr>
<td>ResNet18 own</td>
<td>0.8486 0.8383</td>
<td>0.7186 0.4905</td>
<td>0.7349 0.5462</td>
</tr>
<tr>
<td><strong>ImageNet with CRF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.8605 0.8440</td>
<td>0.7216 0.4880</td>
<td>0.6931 0.4932</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.7248 0.7061</td>
<td>0.7274 0.4974</td>
<td>0.6768 0.4672</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.7824 0.7584</td>
<td>0.7213 0.4868</td>
<td>0.6798 0.4700</td>
</tr>
<tr>
<td><strong>SimCLR with CRF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet18 public</td>
<td>0.8601 0.8548</td>
<td>0.7377 0.5157</td>
<td>0.7220 0.5398</td>
</tr>
<tr>
<td>ResNet18 own</td>
<td>0.9031 0.8977</td>
<td>0.7388 0.5182</td>
<td>0.7433 0.5535</td>
</tr>
</tbody>
</table>
Fig. 4. Feature space of different pre-training methods. (a) ImageNet feature space. (d) Own SimCLR feature space. (b) Raw example and (c) colored by positive and negative patches. The arrows show the location of patches with different morphological entities.

Fig. 5. Cumulative distribution of cases with DSC equal to DSC after four corrections. Already with the initial seeds, 20% of all cases reached DSC equal to DSC after four corrections. Note that the DSC after four corrections is not necessarily 1.0 but values vary as presented in Supplementary Figure 2.

performance in the CAMELYON17 dataset and competitive results in both ACDC and DigestPath, where the best performing supervised method achieved $DSC = 0.8372$ and $DSC = 0.8075$, respectively, the lower performance hinting at them being more challenging problems when compared to CAMELYON17.

4. Discussion

In this study, we presented DEPICTER, an interactive histopathology annotation tool based on clustering deep representations generated by self-supervised learning to obtain a patch-wise segmentation.

The real time experiment with the pathologists proved that in a short time the interaction with the UMAP reduced space enabled accurate segmentation. In the beginning, the pathologists followed the workflow of first selecting the patches on the image and then drawing around the highlighted areas. However, interestingly, once they understood how at times the positive patches were tightly packed in a specific area of the feature space, they started building trust and first drawing in the feature space and then checking the results on the image. To our knowledge, this is the first time a guided interactive tumor annotation tool is evaluated in this way, providing insights into time, accuracy and amount of interactions. Similar concepts were used by Miao et al. [34] or Zheng et al. [49], understanding that similar patches would have similar representations and thus could be used for retrieving similar looking image regions.

The ablation studies showed that using CRF for post-processing provided marginal improvement in the performance and was rather unpredictable, most likely due to using UMAP features as one of the kernels. Additionally, the smallest tested network, ResNet18 generally provided the best results for all experiments, particularly on CAMELYON17 and DigestPath. Even if the results are not better but only comparable for ACDC, one can argue that the time performance from using a lighter network makes it preferable. The balance between the depth of the layer and the amount of convolutions the image undergoes probably explains the difference in performance. Using the last layer before prediction can be considered a rather advanced layer in terms of abstraction and thus a deeper network could not pick the general morphological differences needed for the DEPICTER to work, as also argued in Guérin et al. [50].

We also found that the models pretrained using histopathology outperformed the ones pretrained on ImageNet, which consists of natural images. This was previously shown by Chen and Krishnan [30], where extracted representations were used as input for a subsequent classification network. In our case, we do not use a subsequent classification network, but instead cluster the representations directly. This lets us observe how the features from the models behave, without mixing the effect of the features with learning low order statistics from the data [51].

Considering that a given user might not be able to pretrain the network on their own data, the SimCLR public results are really promising, as they could use publicly available weights pretrained on histopathology imaging to achieve substantial results without the need of having a full dataset on which to pretrain.

As for the model with ImageNet weights, we showed that coarse differences in morphology can be found, which is why these features are still used in many downstream applications [52]. Even though
the seeded iterative clustering approach needs a better defined feature space, for the feature space gating functionality we propose, ImageNet pretrained networks are still able to produce acceptable regions with added effort by the user.

The differences in performance across datasets could be explained both by the difficulty of the problem, as seen by the different performance of the winners of each challenge, and the size of the datasets. Because the CAMELYON17 dataset contained many samples with microscopic lesions that only cover a few patches where the performance of DEPICTER would be biased, it ended up with 17 slides, whereas we could use 104 and 196 images form ACDC and DigestPath, respectively. This meant that training the self-supervised model for the same amount of epochs for all the datasets, the one with fewer images could be allowed to learn more specific representations. Additionally, lower performance in the slides was mainly due to small and sparse lesions that were too close in morphological appearance to benign regions.

4.1. Limitations

Precomputing the feature space leverages the power of deep learning while having a fast and interactive tool, but it also renders DEPICTER dependent on the generated embedding space. If the pathologist chooses a seed that is misplaced in the feature space, it can affect the seeded iterative clustering label propagation or their own manual gating. On a related note, the fact that the size of the patches can cover more than one morphological entity, can confuse the method and again miss-position a patch containing cancer close to benign patches if most of the morphology is benign.

Both these problems could be in principle solved sacrificing time during the precomputing stage [53] by (1) using more powerful self-supervised learning approaches trained for longer epochs or more augmentations and (2) extracting smaller or overlapping patches at higher resolutions.

4.2. Future lines of work

In our previous work, we found that deep morphological features of the kind extracted in this work are correlated with spatial transcriptomes profiles [54]. This could mean that, in the same way we are now segmenting in morphologically relevant compartments, one could obtain genetically relevant ones by interacting with the feature space.

We also envision DEPICTER being used for troubleshooting and enhancing interpretability of deep learning models by analyzing what they are internally learning. As an example, after training a cancer grading model, one could use it to extract features which then are the input to DEPICTER for exploring what are the different morphological entities learned by the model and if they are relevant for the problem. This could be considered an intrinsic qualitative interpretability metric to diagnose potential shortcomings and biases.

The pathologist guided patch-level classification could be used to discard negative patches and reduce the false positive rate of a downstream pixel level classification as proposed in other approaches [48].

Finally, in a more automated scenario, the seeded iterative clustering approach can be also used when the seeds are initialized using auxiliary tasks, such as guided attention.

5. Conclusion

In conclusion, we introduced DEPICTER, an open-source interactive and web-based tool for annotating histopathology WSI. We evaluated DEPICTER both with expert pathologists and using simulations and obtained accurate results for patch-level WSI-wide cancer segmentation in three benchmarks.

CRediT authorship contribution statement

Eduard Chelebian: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis. Christophe Avenel: Writing – review & editing, Visualization, Supervision, Software, Conceptualization. Francesco Giompi: Writing – review & editing, Supervision, Project administration, Conceptualization. Carolina Wählby: Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization.

Declaration of competing interest

C.W. is on the advisory board of Navinci Diagnostics, Sweden. F.C. was Chair of the Scientific and Medical Advisory Board of TRIBVN Healthcare, France, and received advisory board fees from TRIBVN Healthcare, France in the last five years. F.C is shareholder of Aiosyn BV, the Netherlands.

Acknowledgments

This research was funded by the European Research Council via ERC Consolidator grant CoG 682810 and Technology Development project from SciLifeLab to C.W. We are thankful to Patrick Micke, Kevin Sandeman and Max Backman for testing DEPICTER and providing valuable feedback.

Appendix A. Supplementary data

Supplementary Table 1: Average results for feature gating annotation by the pathologists.
Supplementary Figure 1: Examples of discarded cases for ACDC, CAMELYON and DigestPath.
Supplementary Figure 2: Distribution of the DSC results using the best model.
Supplementary Figure 3: Distribution of interactions-to-DSC by the pathologists.
Supplementary Figure 4: Comparison with QuPath pixel classifier

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compbiomed.2024.108026.

References


Table 4

Comparison with the challenge winners for the binary segmentation tasks.

<table>
<thead>
<tr>
<th>Challenge winner</th>
<th>Ours$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAMELYON17</td>
<td>$\kappa = 0.8993$</td>
</tr>
<tr>
<td>ACDC</td>
<td>DSC = 0.8372</td>
</tr>
<tr>
<td>DigestPath</td>
<td>DSC = 0.8075</td>
</tr>
</tbody>
</table>

$^a$ Please note that this is only an orientative comparison, as we explained, the data in the datasets is not the same.
E. Chelebian et al.

Computers in Biology and Medicine 170 (2024) 108026


