Seeded iterative clustering for histology region identification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Annotations are necessary to develop computer vision algorithms for histopathology, but dense annotations at a high resolution are often time-consuming to make. Deep learning models for segmentation are a way to alleviate the process, but require large amounts of training data, training times and computing power. To address these issues, we present *seeded iterative clustering* to produce a coarse segmentation densely and at the whole slide level. The algorithm uses precomputed representations as the clustering space and a limited amount of sparse interactive annotations as seeds to iteratively classify image patches. We obtain a fast and effective way of generating dense annotations for whole slide images and a framework that allows the comparison of neural network latent representations in the context of transfer learning.

1 Introduction

Deep learning segmentation models are already being used for easing the burden of cancer histopathology whole slide image (WSI) annotation [22]. However, supervised training requires annotated images, often long training times and computational resources such as graphical processing units (GPU). To overcome these bottlenecks, approaches such as active learning [20] are gaining popularity, as they reduce the time needed to perform the segmentation while retaining high performance via expert supervision. One strategy is to have pathologists annotate a small part of the image, either by clicking or scribbling around the regions of interest. Then, features are extracted from those sparsely annotated regions and the dense segmentation masks are obtained by retraining segmentation models. This strategy, however, requires both interactive and GPU support at run time.

For this proof-of-concept, we instead propose seeded interactive clustering (SIC) to decouple the problem by (1) precomputing the latent network representations, which does not require a GPU at run time so it can be done remotely in a computing cluster without interaction and, once the embeddings are obtained, (2) perform interactive clustering locally that does not involve any further training and is relatively cheaper computationally.

1.1 Previous work

Different authors have approached this problem by creating interactive tools based on extracting classical features [1, 11] or on deep learning approaches [12, 14, 16]. Some of these approaches are intended for cell detection on smaller images, and therefore are harder to scale to bigger WSI. The approaches that are meant for digital pathology usually require a GPU while performing the interactive annotations, which may not always be available.

Submitted to Learning Meaningful Representations of Life Workshop, NeurIPS 2022. Do not distribute.

The re-purposing of convolutional neural networks (CNN) as feature extractors was proposed very early after the first models trained on large datasets were made avaialable [8, 18, 21, 23]. Since then, using model architectures such as ResNet-50 [10] trained on ImageNet has become common practice for extracting features and performing transfer learning, even if some authors argue that these models may not capture all the relevant histology features [4].

Recent work also proposes using self-supervised learning approaches for obtaining the embeddings [5, 6], under the assumption that the feature space would be more representative of the tissue differences. However comparing different embeddings is not a trivial task and is influenced by many factors [13, 17].

Finally, image-set clustering is a field where all these concepts, including the choice of architecture and layer for feature extraction, or the clustering algorithms, has been extensively studied [9].

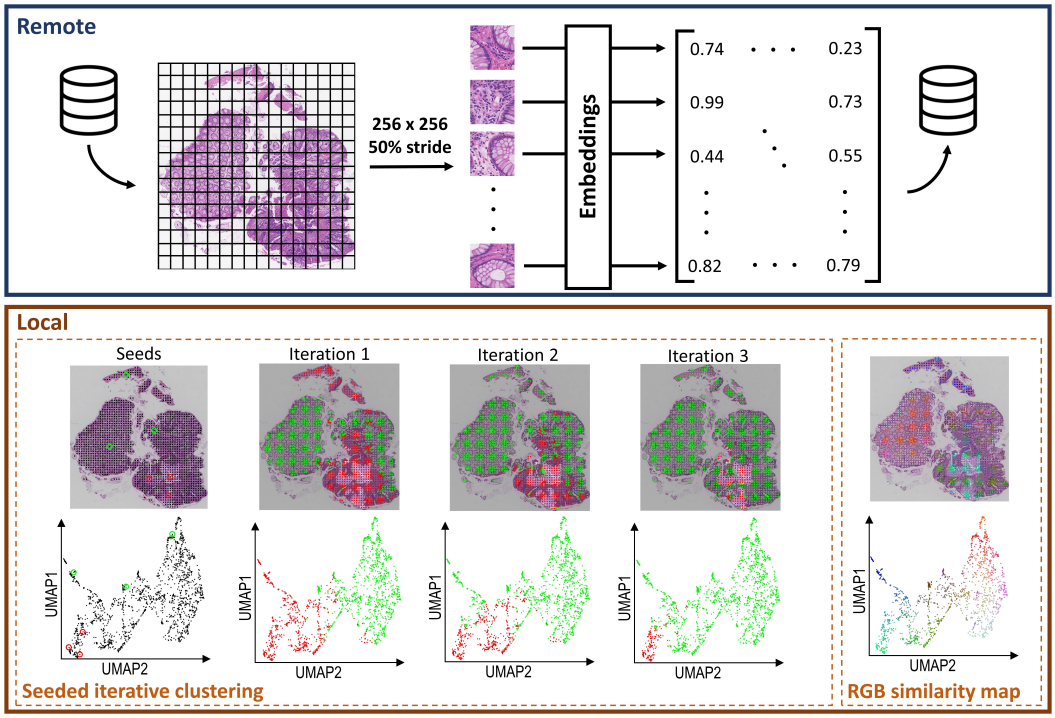

Figure 1: Visualization of the seeded interactive clustering process. The *Remote* box illustrates how tissue patch embeddings are extracted. The *Local* box, from left to right: the top row shows a tissue sample, where seeds (red and green markers) are manually placed on the tissue image. The bottom row shows the embedding space location of the corresponding tissue patches, where black dots are patches not yet assigned to a category. As the SIC iterations continue, the patches are re-clustered according to embedding similarity. Far right: dots representing patches in UMAP space are colored according to their position in RGB color space, and re-mapped to their coordinates in tissue space. Note that the clustering is made on the full embedding space and the reduced space is only intended for visualization purposes. Visualizations created using TissUUmaps [19].

2 Method

2.1 Seeded iterative clustering

The proposed pipeline is presented in Figure 1 and has two parts: one remote and one local. The first remote part of the process starts by dividing the image into smaller patches. The size of the patches, the stride between them and the resolution of choice defines the granularity of the final tissue clustering. In this case, we decided to extract 256×256 patches with 50% overlap at $20\times$ resolution.

Next, each of the patches are embedded using the pretrained model of choice by extracting the last layer latent representation. Finally, we store the matrix of $number\ of\ patches \times embedding\ size$ together with the center coordinates of each patch to represent the complete WSI.

The second part of the pipeline can be run locally. The seeded iterative clustering (SIC) approach, presented in Algorithm 1, divides the previously generated patch embedding space in a binary fashion, guided by small seeds -or sparse annotations- selected by the user in representative regions. Then seeded clustering [2] is performed iteratively until a score relative to the sparse annotations does not improve. We used F1-score, but any classification metric could be used to optimize the algorithm. The seeded KMeans algorithm only differs from the original KMeans in the first assignment of cluster centers, which are not initialized randomly but based on the seeds. An additional step of conventional KMeans is added as it had shown to improve the performance when the tumor region of the WSI was not big enough.

Algorithm 1 Seeded Iterative Clustering

```
X ← embeddings                                          ▷ Embeddings of patches
y ← seeds                                               ▷ Sparse annotations
score_prev ← 0                                          ▷ Initialize score
labels ← SeededKMeans(n_clusters = 2).fit(X, y)         ▷ Initialize labels
while score ≤ score_prev do                             ▷ Run until defined score goes down
    X ← embeddings[labels == 1]                         ▷ Select data only in the postive class
    y ← seeds[labels == 1]
    if {0, 1} ⊂ y then                                  ▷ If there are still annotations of both classes
        labels ← SeededKMeans(n_clusters = 2).fit(X, y) ▷ Re-cluster positive cluster
    else
        if {1} ⊂ y then                                 ▷ If there are only negative annotations
            labels ← KMeans(n_clusters = 2).fit(X)      ▷ Perform conventional clustering
        end if
    end if
    score ← metric(y, label)                            ▷ Compare only sparse annotations
end while
```

2.2 Feature space visualization

Inspired by the spatial-omics field, an intuitive way of visualizing the embeddings is by reducing the dimensionality of the space to three dimensions using Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [15]. This 3D space can be mapped to the RGB space and visualized directly on top of the images as colored dots at the center of each patch in the image, as previously presented by Chelebian et al. [3] and shown in Figure 1 (bottom-right).

In the UMAP space, it is possible to manually select regions that are clustered together and see if they correspond to biologically relevant regions on the image. This approach however, is highly dependent on the dimensionality reduction algorithm and should be used only for qualitative visualization purposes.

3 Experimental results

3.1 Dataset

For this proof-of-concept we used the DigestPath challenge cancer segmentation dataset [7]. It is an open database including 250 images of tissue from 93 positive WSI (i.e. with pixel-level annotations) slides in JPEG format which were used for the experimental results. The additional 410 images from 231 WSI negative slides were only used together with the positive slides when self-supervised pretraining was performed. All WSI are hematoxylin-eosin (H&E) stained and scanned at 20× resolution.

3.2 Experiments

In order to explore how different embedding spaces would affect the clustering performance, we chose four different feature extraction frameworks: (1) ResNet-18 pretrained on ImageNet (ResNet18), (2) ResNet-50 pretrained on ImageNet (ResNet50), (3) ResNet-18 pretrained on the dataset using SimCLR self-supervision [5] (SimCLR) and (4) ResNet-18 pretrained on 57 histopathology datasets using SimCLR self-supervision [6] (HistoSSL).

To mimic the complete workflow where the pathologist would select small benign and cancerous regions, we chose to experiment by randomly selecting 1, 3 and 5 patches that were located inside the benign and cancerous masks, respectively. Knowing that this would probably underestimate the results, as the seeds may not be representative of what a pathologist would choose, we report the F1-score for the patch classification task of the best run per slide in a 5-fold experiment, to simulate the selection an expert would make. The ground-truth coordinates are generated by selecting the patches that are at least 50% in a cancer region.

3.3 Results

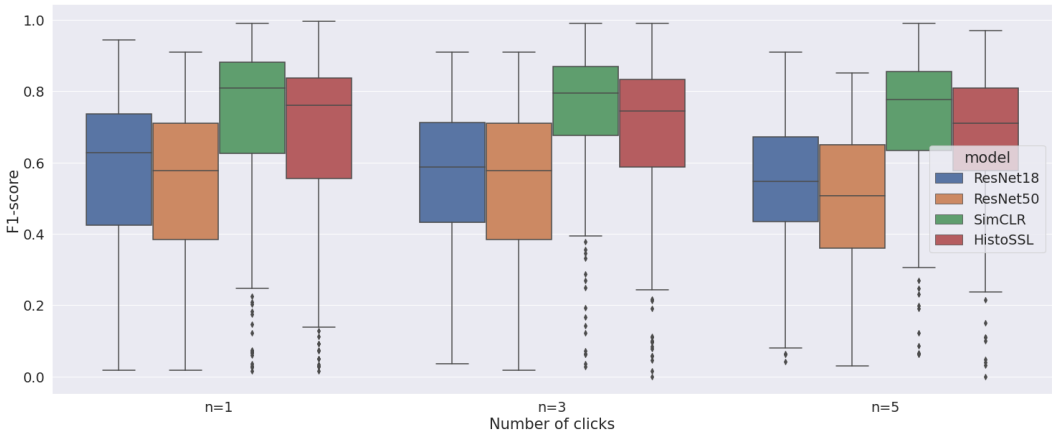

Figure 2: F1-score for the different models with varying number of simulated clicked patches.

The results are summarized in Figure 2. The highest per-slide F1-score for the ImageNet pretrained models are very similar regardless of the architecure, 0.57 ± 0.22 for ResNet18 and 0.54 ± 0.21 for ResNet50, both lower than the self-supervised approaches. There seems to be some benefit for pretraining (SimCLR) on the dataset at hand (0.74 ± 0.23), but this may not be always possible, so it is very promising that a publicly available model (HistoSSL) also achieves higher results (0.68 ± 0.22). The method itself shows high classification variability, although the worse performing slides are usually the ones that have low positive patch ratio, that is, when the lesion is very small compared to the whole tissue, such as Figure 3 (upper right). There is no clear difference from having more seeds, except when it comes to the standard deviation. As they are only used for initialization it seems that if the feature space is descriptive enough, adequately selected patches are sufficient for a good clustering result.

4 Discussion

We presented seeded iterative clustering (SIC) as a way of leveraging latent representations of neural networks to speed-up the time-consuming process of manual annotation in histopathology. From an annotation tool point of view, the local clustering module takes seconds, even on a normal computer. The embedding time depends on the amount of patches or coordinates one wants. However, contrary to previous work, this can be run remotely without the pathologist interaction and does not need to be

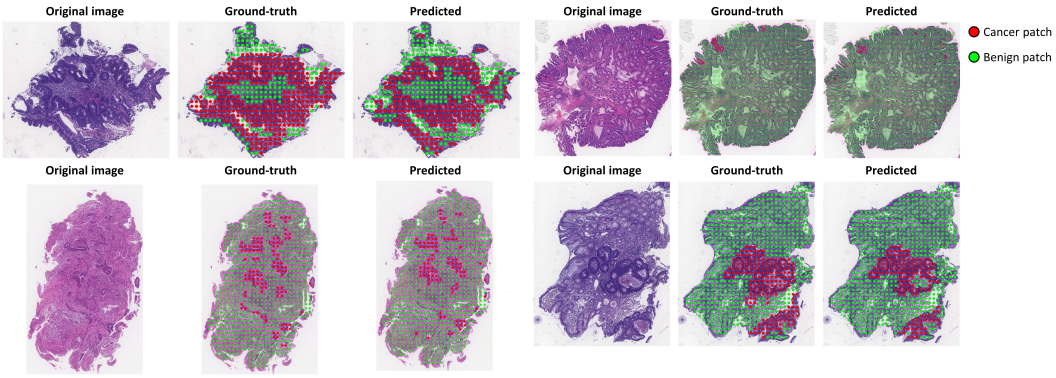

Figure 3: Examples of predictions using seeded iterative clustering. Visualizations created using TissUUmaps [19].

recalculated at any point. Additionally, performing the clustering per-slide is more robust to domain shift artifacts.

From a technical point of view, it allowed to compare the representations from different networks. When performing transfer learning, there is a separate effect of the feature reuse and learning low-level statistics from the data [17]. Retraining, then, may not be ideal to explore the quality of the features. This is precisely why we find that ImageNet pretrained ResNet off-the-shelf are not comparable to using self-supervised pretraining with SimCLR.

Work in progress includes validating this approach on other datasets and developing it into a full interactive tool in which pathologists select a handful of patches and the method proposes clusters for the rest. Additionally, we are currently working on solving the inferior performance for the slides with smaller tumor region by devising a better sampling scheme. Finally, extending this approach in a multi-class problem (e.g. cancer, benign and stroma regions, instead of binary), would provide further insights into how the feature space generated by the different models as well as increase the usability of the tool.

Broader Impact

The uncoupling of the feature extraction and clustering modules could allow for the remote generation of embeddings that can then be shared with the community. This would allow for groups with less computational resources to work with the data and circumvent the potential privacy issues for sharing medical images, sharing instead the latent representation matrices per coordinate.

References

[1] Arzt, M., Deschamps, J., Schmied, C., Pietzsch, T., Schmidt, D., Tomancak, P., Haase, R., & Jug, F. (2022). Labkit: Labeling and segmentation toolkit for big image data. *Frontiers in Computer Science*, 10.

[2] Basu, S., Banerjee, A., & Mooney, R. (2002). Semi-supervised clustering by seeding. *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*.

[3] Chelebian, E., Avenel, C., Kartasalo, K., Marklund, M., Tanoglidi, A., Mirtti, T., Colling, R., Erickson, A., Lamb, A. D., Lundeberg, J., et al. (2021). Morphological features extracted by ai associated with spatial transcriptomics in prostate cancer. *Cancers*, *13*(19), 4837.

[4] Chen, R. J., & Krishnan, R. G. (2022). Self-supervised vision transformers learn visual concepts in histopathology. *arXiv preprint arXiv:2203.00585*.

[5] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, 1597–1607.

[6] Ciga, O., Xu, T., & Martel, A. L. (2022). Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, *7*, 100198.

[7] Da, Q., Huang, X., Li, Z., Zuo, Y., Zhang, C., Liu, J., Chen, W., Li, J., Xu, D., Hu, Z., et al. (2022). Digestpath: A benchmark dataset with challenge review for the pathological detection and segmentation of digestive-system. *Medical Image Analysis*, 102485.

[8] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. *International conference on machine learning*, 647–655.

[9] Guérin, J., Thiery, S., Nyiri, E., Gibaru, O., & Boots, B. (2021). Combining pretrained cnn feature extractors to enhance clustering of complex natural images. *Neurocomputing*, *423*, 551–571.

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

[11] Hollandi, R., Diósdi, Á., Hollandi, G., Moshkov, N., & Horváth, P. (2020). Annotatorj: An imagej plugin to ease hand annotation of cellular compartments. *Molecular biology of the cell*, *31*(20), 2179–2186.

[12] Jaber, M. I., Song, B., Beziaeva, L., Szeto, C. W., Spilman, P., Yang, P., & Soon-Shiong, P. (2021). A deep learning-based iterative digital pathology annotation tool. *bioRxiv*.

[13] Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. *International Conference on Machine Learning*, 3519–3529.

[14] Lindvall, M., Sanner, A., Petré, F., Lindman, K., Treanor, D., Lundstrbm, C., & Ldwgren, J. (2020). Tissuewand, a rapid histopathology annotation tool. *Journal of pathology informatics*, *11*(1), 27.

[15] McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

[16] Miao, R., Toth, R., Zhou, Y., Madabhushi, A., & Janowczyk, A. (2021). Quick annotator: An open-source digital pathology based rapid image annotation tool. *The Journal of Pathology: Clinical Research*, *7*(6), 542–547.

[17] Neyshabur, B., Sedghi, H., & Zhang, C. (2020). What is being transferred in transfer learning? *Advances in neural information processing systems*, *33*, 512–523.

[18] Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1717–1724.

[19] Pielawski, N., Andersson, A., Avenel, C., Behanova, A., Chelebian, E., Klemm, A., Nysjö, F., Solorzano, L., & Wählby, C. (2022). Tissuumaps 3: Interactive visualization and quality assessment of large-scale spatial omics data. *bioRxiv*.

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

[21] Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). Cnn features off-the-shelf: An astounding baseline for recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 806–813.

[22] Van der Laak, J., Litjens, G., & Ciompi, F. (2021). Deep learning in histopathology: The path to the clinic. *Nature medicine*, *27*(5), 775–784.

[23] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European conference on computer vision*, 818–833.