Efficient Self-Supervision using Patch-based Contrastive Learning for Histopathology Image Segmentation

Nicklas Boserup[1] and Raghavendra Selvan[*1,2]

[1]Dept. of Computer Science, University of Copenhagen
[2]Dept. of Neuroscience, University of Copenhagen

Abstract

Learning discriminative representations of unlabelled data is a challenging task. Contrastive self-supervised learning provides a framework to learn meaningful representations using learned notions of similarity measures from simple pretext tasks. In this work, we propose a simple and efficient framework for self-supervised image segmentation using contrastive learning on image patches, without using explicit pretext tasks or any further labeled fine-tuning. A fully convolutional neural network (FCNN) is trained in a self-supervised manner to discern features in the input images and obtain confidence maps which capture the network's belief about the objects belonging to the same class. Positive- and negative- patches are sampled based on the average entropy in the confidence maps for contrastive learning. Convergence is assumed when the information separation between the positive patches is small, and the positive-negative pairs is large. The proposed model only consists of a simple FCNN with 10.8k parameters and requires about 5 minutes to converge on the high resolution microscopy datasets, which is orders of magnitude smaller than the relevant self-supervised methods to attain similar performance. We evaluate the proposed method for the task of segmenting nuclei from two histopathology datasets, and show comparable performance with relevant self-supervised and supervised methods. [1]

---

*Corresponding Author: raghav@di.ku.dk
[1]Source code available at: `https://github.com/nickeopti/bach-contrastive-segmentation`

1 Introduction

Learning task specific representations without any – or with limited – labelled data continues to be an elusive goal of machine learning. Recent advancements in contrastive learning and self-supervised learning have shown promising results in obtaining discriminative representations of the data which can be useful for downstream applications such as image classification [2], object detection [9] and speech recognition [1]. Contrastive self-supervised learning (CSL) has been successfully used as a form of pre-training to reduce the dependence on labeled data for more complex tasks such as image segmentation [10]. Most self-supervised methods rely on pretext tasks for training them [6, 8]. Designing relevant pretext tasks can be challenging and even if a useful pretext task is obtained they may not easily generalise across datasets [5].

In this work, we present a self-supervised learning framework that contrastively learns an object detection model for segmenting nuclei in histopathology images. The model comprises a fully convolutional neural network (FCNN) that predicts one confidence map per output channel, which captures the confidence of each pixel belonging to a particular object class. The FCNN is contrastively trained using smaller positive- and negative patches stochastically sampled from the images. Patches within a training batch are sampled from an entropy based distribution, where the entropy is based on the patch-level confidence scores. The intuition behind the entropy-based sampling is to obtain positive patches that contain similar information and negative patches that contain contrasting information, with respect to features that can discriminate between objects of different classes. Through iterative training we are able to improve the information separation between the positive- and negative patches, resulting in an object
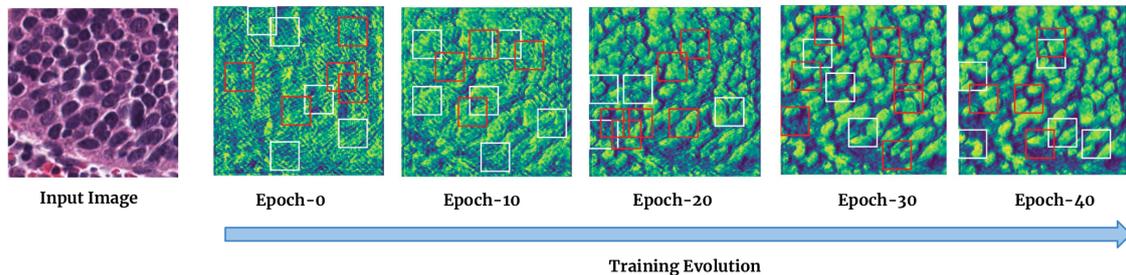
Figure 1: Overview of the training evolution of the proposed contrastive self-supervised learning model. At each of the five illustrated epochs, predicted confidence map (from which segmentation masks are derived by thresholding) for a single validation set image is shown, along with the patches sampled as positive (white squares) and negative (red squares).

detection model which can be used for segmentation.

We use a simple FCNN with 10.8k tunable parameters which converges in about 5 minutes on a stand-alone GPU workstation. Experimental evaluation on two histopathology datasets [4, 3] show that our efficient, contrastive self-supervised learning method obtains performance comparable to relevant supervised and self-supervised baseline methods, using only a fraction of the compute time and resources. An illustration of how the confidence map evolves during the training process is illustrated in Figure 4.

2 Method

Segmentation is fundamentally the task of partitioning an image into areas of interest and a background class. Assuming that the areas of interest within the same class are similar in some feature space according to a *similarity measure*, we present our contrastive learning framework that results in an unsupervised segmentation model. A high level overview of the proposed framework is illustrated in Figure 2.

2.1 Notation

Consider a batch of M images, with the i'th image represented as the pair (U_i, I_i), where $U_i \subset \mathbb{N}^d$ is a finite set representing the d-dimensional pixel coordinates and I_i is a function $I_i : U_i \to [0, 1]^\lambda$, with $\lambda \in \mathbb{N}$ denoting the number of colours/channels, mapping such pixel coordinates into their respective values.

Next, we introduce the notation for denoting the patch locations sampled from the image i as the tuple (R, i), where $R \subset U_i$ is the set of pixel coordinates of the image patch. Note that the power set $\mathcal{P}(U_i)$

denotes the set of all possible smaller patches that can be sampled from image i. Denoting the set of all such patch tuples across images $\mathcal{X}$, we consider the subset $S \subset \mathcal{X}$ of regular (square) and spatially connected patches in this work.

The proposed framework uses a *confidence network* (Section 2.2), f_θ, parameterised by θ which for a given input image (U_i, I_i) computes the *confidence map*, C_i, i.e., $f_\theta : (U_i, I_i) \in [0, 1]^\lambda \to C_i \in [0, 1]^K$. The k'th coordinate in $C_i^k(u)$ indicates the belief, or confidence, that the pixel $u \in U_i$ in image i belongs to class k, for $k = 1, ..., K$, with $K \geq 1$.

The contrasting of patches depends on a similarity measure (Section 2.4), $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which compares the similarity of two patches. This similarity computation relies on (I_i, I_j) and (C_i, C_j), where i and j are the images from which the two compared patches originate from.

2.2 Confidence Network

The aim of the confidence network, f_θ, is to learn to detect objects in images without any pixel-level supervision, such that areas with high confidence of the k'th class actually belong to a specific type of object in the images. Conversely, objects detected with high confidence of the j'th class, when $k \neq j$, should belong to a different class. The confidence network, f_θ, can be any trainable model which given an input image produces a set of *confidence maps*, describing the confidence that each pixel of the image belongs to a corresponding class. This, for instance, can be implemented as a fully convolutional neural network (FCNN), with multi-class support i.e., with multiple output channels such that each channel would correspond to a different output class.

I: Input Image   C: Confidence Map
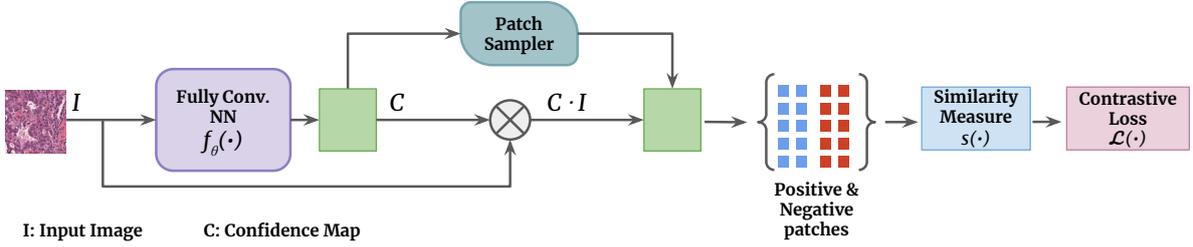
Figure 2: High level overview of the proposed patch-based contrastive self-supervision method. The input color image, **I**, is input to the fully convolutional neural network, $f_\theta(\cdot)$, to obtain a pixel level foreground confidence map, **C**. The confidence map is used by the *patch sampler* which returns locations of positive and negative patches. The patches are then obtained from the attended image, **C·I**, to obtain the inter- and intra- class similarity measures, $s(\cdot)$. A contrastive loss, $\mathcal{L}(\cdot)$, is computed based on these similarities which is then back-propagated through the entire pipeline. The trainable weights, θ, are tuned until the confidence map, **C**, corresponds to segmentation masks of interest.

The confidence network is trained to discriminate between objects that could belong to different classes by contrasting *patches* of images against each other. Obtaining meaningful positive- and negative patches for the contrastive learning is the foundational problem in this framework. We next describe a novel approach to mine for such contrastive patches for self-supervision of the confidence network.

2.3   Entropy-based Patch Sampler

For each output class $k=1,...,K$, patches *believed* to contain (a part of) an object belonging to a specific class are treated as *positive* patches, whereas patches *believed* to not contain (a part of) an object of that class are treated as *negative* patches. Appropriate sampling of such positive- and negative patches is of vital importance, as the optimisation of the confidence network, f_θ, is performed according to the contrastive loss computed based on those patches. Improved object detection shall ideally correlate with positive and negative sampled patches being more distinct. Effectively, the appropriate sampling of patches becomes a form of pretext/auxiliary task for training the confidence network, f_θ.

Assume a set of candidate patches, $S \subset \mathcal{X}$. For each patch $(R,i) \in S$ with $|R|$ pixels and for each class k, the average patch confidence is computed as:

$$A_k(R,i) := \frac{1}{|R|} \sum_{u \in R} C_i^k(u) \in [0,1]. \qquad (1)$$

Notice that the higher the average patch confidence,

the stronger the belief that the patch contains the type of object belonging to class k.

Recall that C_i has range $[0,1]^K$, that is, for each class, $k=1,...,K$, a $C_i^k(u)$ value closer to 1 indicates a higher probability of u belonging to the k'th class. Likewise, a value closer to 0 indicates that u likely does not belong the k'th class, whereas values in between indicate varying degrees of (un)certainty in either direction. Using this intuition about the confidence maps, we model a Bernoulli distributed random variable, $X(u)$, with confidence value as its parameter, denoted as $X(u) \sim \mathrm{Bern}(C_i^k(u))$. Using this, we can now define the average patch entropy as

$$B_k(R,i) := \frac{1}{|R|} \sum_{u \in R} H(X(u)) \qquad (2)$$

where

$$H(X) = - \sum_{x \in \{0,1\}} \mathbb{P}(X(u)=x) \log_2 \mathbb{P}(X(u)=x) \qquad (3)$$

$$= -C_i^k(u) \log_2 C_i^k(u) - (1-C_i^k(u)) \log_2(1-C_i^k(u)), \qquad (4)$$

which is the entropy of the Bernoulli distributed random variable $X(u)$. The motivation for this choice of Bernoulli random variable is so that good choices of positive- and negative patches, according to the contrastive loss, would correlate with higher certainty from the confidence network. Therefore sampling a set $W_k \subset S$ of n patches according to the unnormalised distribution

$$1 - B_k(R,i)$$

for each class k ensures a stochastic correlation between the confidence map and the appropriateness

3

of patch sampling. Finally, partitioning of these W_k into sets of positive samples, P_k and negative samples N_k is performed according to

$$[(R,i) \in P_k] \sim \text{Bern}(A_k(R,i))$$

for each patch $(R,i) \in W_k$, such that patches with high confidence of belonging to class k are likely to be selected as positive for class k.

2.4 Similarity Measures

Ideally, the similarity measure, $s \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, should yield higher values when two input patches are more similar. In this work, for any two patches $(R_1,i),(R_2,j) \in S$, the similarity measure performs the comparison on the image scaled by the corresponding confidence map, given by the following pixel-wise products

$$F_i(u) := I_i(u)C_i(u), \quad u \in R_1 \tag{5}$$

$$F_j(v) := I_j(v)C_j(v), \quad v \in R_2 \tag{6}$$

We employ two pixel-level[2] similarity measures; mean squared error, or difference, (MSE) and mean cross-entropy (CE), and treat them as a model hyperparameter and report the best performing measures for each of the datasets. We envision density based similarity measures to be desirable, but in our limited experiments they performed poorly.

Backpropagation: The scaling of the input image with the confidence map also serves the purpose of connecting the gradients between the sampling step and the confidence network for backpropagating the contrastive loss to optimise the confidence network.

2.5 Contrastive Loss

Within a channel k of the confidence map, we seek to maximise similarity between positive patches while minimising the similarity between positive- and negative- patches. This is facilitated using the intra-channel contrastive loss

$$\mathcal{L}_{\text{intra}} = -\sum_{k=1}^{K} \frac{1}{|P_k|^2} \sum_{r,p \in P_k} \log \frac{\exp(s(r,p))}{\sum_{n \in N_k} \exp(s(r,n))}. \tag{7}$$

This contrastive loss, inspired by SimCLR [2], rewards f_θ for learning to detect a single feature in the images for each class in the confidence map, when patches are sampled appropriately. However, this lacks any mechanism for enforcing the individual classes to learn distinct features. In a multi-class scenario, each of the classes shall ideally detect different features. This is achieved by introducing an inter-channel contrastive loss

$$\mathcal{L}_{\text{inter}} = -\sum_{k=1}^{K} \frac{1}{|P_k|^2} \sum_{r,p_k \in P_k} \frac{\exp(s(r,p_k))}{\sum_{i=1,[i \neq k]}^{K} \sum_{p \in P_i} \exp(s(r,p))}. \tag{8}$$

Finally, the combined loss is defined as

$$\mathcal{L} = \mathcal{L}_{\text{intra}} + \mathcal{L}_{\text{inter}}. \tag{9}$$

3 Data & Experiments

3.1 Data

As the proposed CSL framework is trained exclusively using unlabelled images, without exposure to the labels during training, we can also use the labels from the training set for validation of the segmentation performance. That is, we train the weights of the FCNN without using any labels but can employ the labels for model selection.

MoNuSeg: This dataset [4][3] from MICCAI 2018 contains 37 training images at 1000×1000 pixels resolution, obtained at $40\times$ magnification, as well as 14 testing images at the same resolution. Dense annotations of all nuclei in all images are provided.

CoNSeP: This dataset [3][4] contains 27 training images at 1000×1000 pixels resolution, obtained at $40\times$ magnification, as well as 14 similar testing images. There are dense annotations of all nuclei in all images.

3.2 Experimental Set-up

Baseline Methods: The simplest supervised baseline method is to obtain the optimal intensity threshold using the training images. The images are converted to grey-scale and an optimal threshold to obtain binary segmentation is obtained. Additionally,

---

[2]Patches are compared pixel-by-pixel; formally, given a bijection g between elements $u \in R_1$ and $v \in R_2$, patches are compared by $s(F_i(u),F_j(g(v)))$. In practice, we compare pixels at same relative positions within the equisized patches.

[3]MoNuSeg data is available at `https://monuseg.grand-challenge.org/Data/`

[4]CoNSeP data is available at `https://warwick.ac.uk/fac/cross_fac/tia/data/hovernet/`

# Patches	5	**10**	15	20	25
Dice	0.6643	0.7071	0.6682	0.6986	0.6059
Patch Size	20^2	30^2	$\mathbf{50^2}$	80^2	120^2
Dice	0.5063	0.6562	0.6980	0.6606	0.6871

Table 1: The median validation Dice score over three runs on MoNuSeg is used to select number of patches and patch sizes (shown in boldface).

Methods	MoNuSeg	CoNSeP	# Par.	Time
Int. Thresh.	0.5823	0.5700	1	<1m
CNN	0.7666	0.7375	10.8k	≈ 10m
Scale Pretext	0.6209	0.5139	21.7M	> 12h[6]
Ours: Max.	0.6797	0.6266		
Mean ± s.d	0.62±0.05	0.52±0.06	10.8k	≈ 5m
80 perc.	0.6469	0.5448		

Table 2: Dice scores on test sets of MoNuSeg and CoNSeP datasets for the different methods along with maximum, mean+sd and 80'th percentile scores for our method. Optimal intensity threshold method (Int. Thresh.), supervised CNN baseline, self-supervised method using scale prediction pretext task (Scale Pretext) and the proposed CSL framework. The number of trainable parameters and convergence time per method are also reported.

a CNN of the same architecture as the confidence network, f_θ, is used as a supervised baseline. And finally, these results are also compared to the self-supervised method that uses scale prediction as a pretext [8], which is referred to as the Scale Pretext method. All baselines are evaluated on both mentioned datasets.

Model Hyperparameters: The default configuration uses a confidence network, f_θ, closely inspired by [8]. Entropy based stochastic sampling described in Section 2.3 and $K=4$ classes are used. The patch size of 50×50 pixels and 10 patches from each image in a batch are chosen based on experiments on the MoNuSeg dataset, as reported in Table 1.

The proposed framework was trained for a maximum of 300 epochs. To limit RAM usage, the input images are cropped into 300×300 pixels, where the location is selected uniformly at random as to add some data variance. No other pre-processing nor data augmentation is applied. Batch size of 10 is used for all experiments.

Implementation: The proposed framework is implemented in PyTorch [7], with support to be trained on GPUs; all training has been performed on a system with an NVIDIA GeForce RTX 3060 and intel-i7 processor with 32GB memory.

Experiments: With the objective of comparing the segmentation performance of the proposed CSL framework with the baselines, we perform experiments on each of the datasets described in Section 3.1.

Our CSL framework is initialised with the hyperparameters described in Section 3.2 on each of the datasets and the weights of the confidence network are randomly initialised. A single training epoch consists of contrasting 10 patches of size 50×50 (see Table 1) sampled from one random crop from each training image. At the end of each epoch, the confidence maps obtained from the confidence network are thresholded[5] at $p=0.5$ and the segmentation performance is evaluated using the labels for the training images by computing the Dice score. We limit the training of the models

for a maximum of 300 epochs, as for both datasets we noticed that the learning plateaued around 200 epochs. After training for 300 epochs, the model at epoch with highest validation Dice score is selected for evaluation on the test set. These Dice scores are reported and discussed further in Section 4. Simple post-processing comprising two iterations of morphological opening and closing operations with radius 3 and 1, respectively, are performed on the thresholded segmentation masks; this primarily helps remove some high frequency noise in the predicted segmentation masks.

The proposed CSL framework is trained 10 times with different initialisations on each dataset, in order to illustrate and quantify its inherent variance in segmentation performance.

4   Results

Segmentation performance of our method and the baselines are reported in Table 2 on both the datasets. The first two rows show the performance of two supervised methods and the third row shows the self-supervised Scale Pretext baseline method [8]. We ran ten random repeats of our model for each dataset configuration and report the mean, maximum

---

[5] The confidence maps, as illustrated in Figure 3, are saturated to the extrema (close to 0 or 1); thus the segmentation performance might not be particularly sensitive to the specific choice of the threshold levels.

[6] This could be a large under-estimation of the actual convergence time based on in-house implementation as the actual run-time was not reported in [8].

| Test Image | Ground Truth | Confidence Map | Prediction |
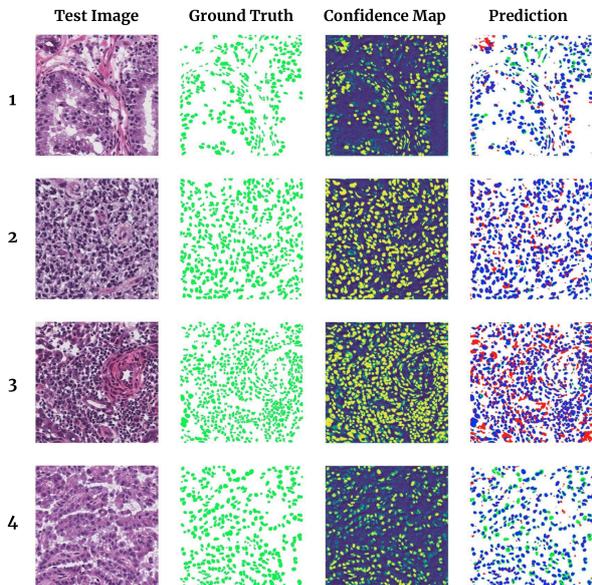
Figure 3: Four test set images from the MoNuSeg dataset. The rightmost column is colour-coded, where blue indicates correct segmentation (true positives), green indicates false negatives, red indicates false positives, and the white background corresponds to true negatives. The predictions are obtained from the confidence map by a 0.5 threshold.

and the 80'th percentile Dice score over these runs. Dice score for the Scale Pretext method is obtained from the paper [8]. In all cases, the supervised CNN method performs better than the self-supervised class of methods, which is to be expected, as no further fine-tuning of these methods with labels is performed.

MoNuSeg: Our method outperforms the supervised optimal threshold method when trained on the MoNuSeg dataset using MSE-similarity. Qualitative results on four randomly selected test set images from the MoNuSeg dataset for the model trained on WSI data are shown in Figure 3.

CoNSeP: A similar performance trend is observed on the CoNSeP dataset, where we notice the mean Dice score of our method with the CE similarity measure performs better than the Scale Pretext method. However, the intensity thresholding method performs better than all self-supervised methods.

The number of parameters of our method (10.8k) is several orders of magnitude fewer than Scale Pretext method (21M), and our method takes only a fraction of

time until convergence (5m) compared to theirs (>12h) as shown in Table 2. This fast convergence time has implications in overcoming the strong influence of the confidence maps obtained at initialisation (epoch-0) which can result in convergence to poor solutions. While some runs result in poor performance – dragging the mean down – most runs yield good performance. We capture this as the 80'th percentile performance over multiple runs, which is better than the mean performance. Thus running the model several times (which is still fast) will likely result in good performing solutions. As a standard practice, we therefore recommend training multiple repeats of our model on any dataset, and selecting the best performing one for inference.

5   Discussions & Conclusions

Confidence maps to segmentation masks: The efficient CSL framework presented in this work outputs multiple confidence maps corresponding to objects believed to belong to the same class. Going from these confidence maps to the segmentation masks by definition requires expert input. In this work, we alleviate this by using the labels for model selection/validation. For other unlabelled datasets, some other form of validation would be required to align the concept of a biomedical image foreground to that of a confidence map of a self-supervised framework.

Self-supervised segmentation performance: The main baseline we compared our method to is the self-supervised Scale Pretext method [8]. The results of this method are not significantly different from ours. This is even with specific preprocessing (stain-normalisation) and elaborate post-processing followed in [8], which are altogether left out in our work. Further, compared to the Scale Pretext method ours is significantly simpler, both in terms of the model complexity, the elaborate regularisation of their objective function, and training time (5m versus 12h).

Mechanism: The aim of the proposed CSL framework is to increase the information separation between positive- and negative- patches. In Figure 4 the learning mechanism is illustrated, where the distribution of average patch confidence and entropy at each of the first 25 epochs of a training process are shown. Conceptually, and in an idealised setting, we would expect the distribution to evolve from being a unimodal distribution centered around 0.5

6
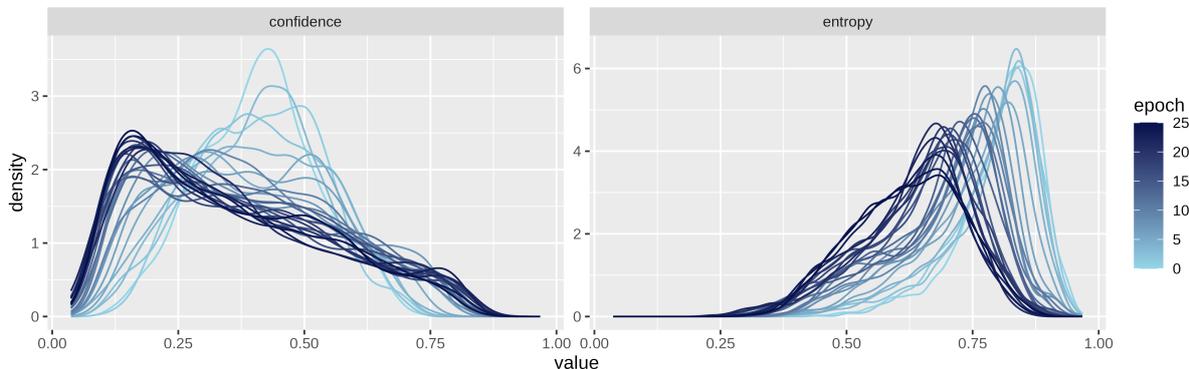
Figure 4: Evolution of confidence network inferences during training process. The two plots illustrate the distribution of average patch confidence and -entropy, respectively, stratified over epoch. This visualises how the distributions change during training; the broadening of the average patch confidence distribution corresponds to an increased information separation. Correspondingly, the average patch entropy decreases. We believe this to be an insight into the mechanism behind the proposed framework.

to a bimodal distribution with peaks close to 0 and 1. While the average patch confidence in Figure 4 does not quite reach a bimodal distribution, it does arguably seem to evolve towards it. We conjecture that this training evolution plot illustrates the fundamental mechanism underlying the proposed CSL framework.

Limitations: The main limitation in the proposed framework coincides with its simplicity — there are no constraints encouraging the model to actually segment the desired objects of interest, potentially resulting in non-useful segmentation maps. Increasing K, the number of classes, may increase the chance of one of them being useful, though this comes at a considerable computational cost; hence the pragmatic choice of $K = 4$ throughout the paper. To further increase the chance of obtaining a useful model, we suggest a standard practice of training the model multiple times, and subsequently selecting the best performing one; the framework is quite sensitive to initialisation, resulting in considerable variance in performance. Fortunately, the 80-percentile scores in Table 2 indicate that the framework often yields useful models, at least on these data sets. And training multiple times is tenable due to the efficiency of the framework.

Finally, we assume that the similarity measure plays an important role in model performance. We experimented with a few of these measures. However, thorough investigations of more refined similarity measures and their influence on multi-class segmentation remains future work.

Conclusions: We presented a self-supervised framework for segmenting nuclei in histopathology data, which uses patch-based contrastive learning. We introduced a novel technique to mine for positive- and negative patches for contrasting, based on the average entropy of the confidence maps. This approach encourages the trainable confidence network to discern objects of different classes, increasing information separation. The resulting method with only 10.8k trainable parameters takes under 5 min. to converge, yielding useful segmentation masks. We foresee interesting research directions for this work that can make it better suited for diverse image data.

Acknowledgements

References

[1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc., 2020. doi: 10.48550/ARXIV.2006.11477.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. doi: 10.48550/ARXIV.2002.05709.

[3] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019. doi: 10.1016/j.media.2019.101563.

[4] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Transactions on Medical Imaging*, 36(7):1550–1560, 2017. doi: 10.1109/TMI.2017.2677499.

[5] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. doi: 10.48550/ARXIV.1912.01991.

[6] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. doi: 10.1007/978-3-319-46466-4_5.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. doi: 10.48550/ARXIV.1912.01703.

[8] M. Sahasrabudhe, S. Christodoulidis, R. Salgado, S. Michiels, S. Loi, F. André, N. Paragios, and M. Vakalopoulou. Self-supervised nuclei segmentation in histopathological images using attention. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 393–402. Springer, 2020. doi: 10.1007/978-3-030-59722-1_38.

[9] J. Xie, X. Zhan, Z. Liu, Y.-S. Ong, and C. C. Loy. Unsupervised object-level representation learning from scene images. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. doi: 10.48550/arXiv.2106.11952.

[10] X. Xie, J. Chen, Y. Li, L. Shen, K. Ma, and Y. Zheng. Instance-aware self-supervised learning for nuclei segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 341–350. Springer, 2020. doi: 10.1007/978-3-030-59722-1_33.