Relieving pixel-wise labeling effort for pathology image segmentation with self-training

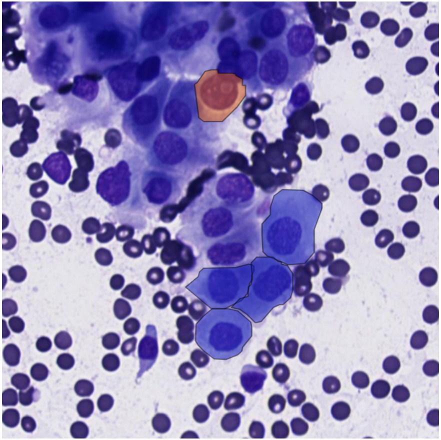
Romain Mormont, Mehdi Testouri, Raphaël Marée & Pierre Geurts

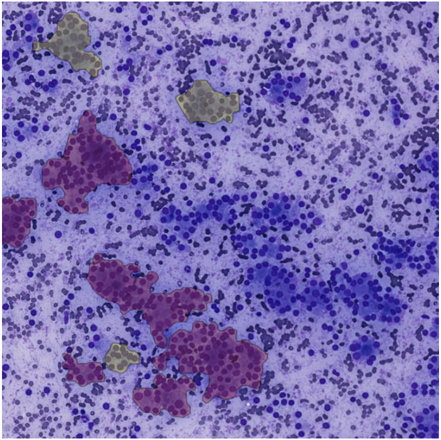
University of Liège, Belgium

October 24, 2022

Thyroid FNAB: an imperfectly-annotated cytology dataset

A sparsely annotated dataset for thyroid nodule malignancy assessment.

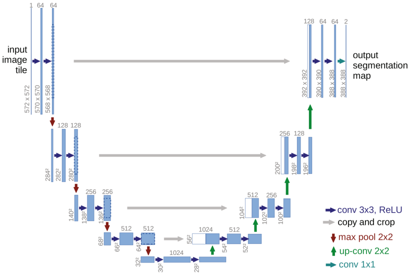




Annotated by the team of Prof. Isabelle Salmon from Erasme hospital (Université Libre de Bruxelles, Belgium).

Using sparsely-labeled data

How to exploit such a sparse/incomplete segmentation dataset in a supervised learning settings ?



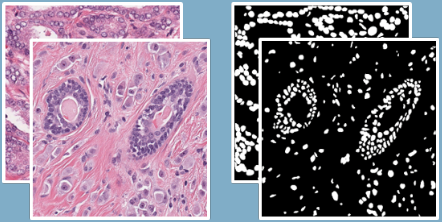
(Ronneberger, Fischer and Brox, 2015)

Our proposal: use the segmentation model being trained to generate the missing information

$\Rightarrow$ **self-training** $\Leftarrow$

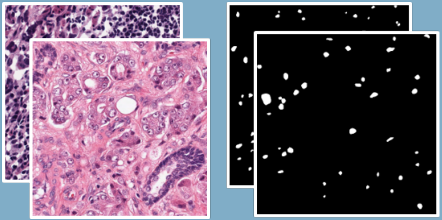
Sparsely-labeled settings

$\mathcal{D}_l$ - exhaustively-labeled set



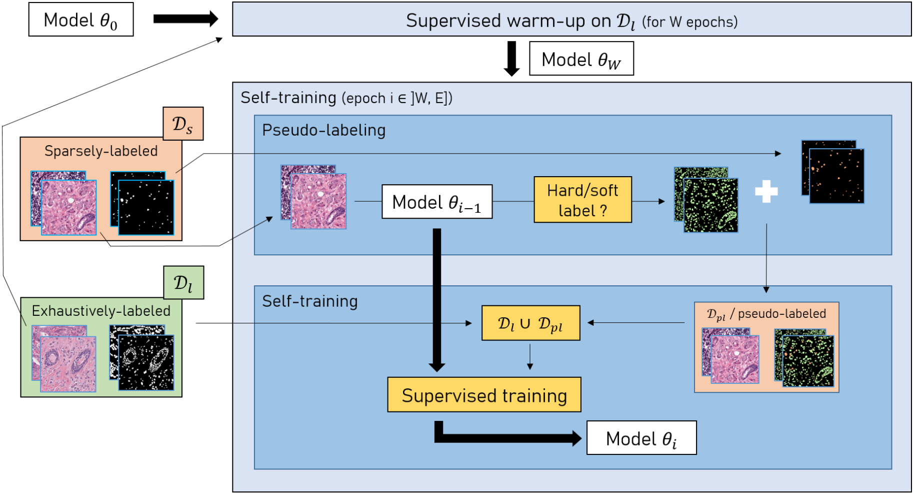
n_l images and masks. All pixels have a 0 (background) or 1 (foreground) label.

$\mathcal{D}_s$ - sparsely-labeled set



n_s images and masks. Unlabeled pixels have label 0 (background) and labeled pixels are exclusively foreground.

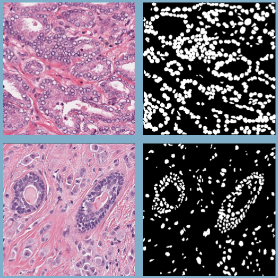
Our self-training algorithm



Experiments

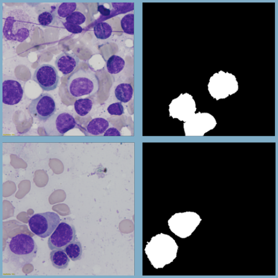
3 public datasets

MoNuSeg



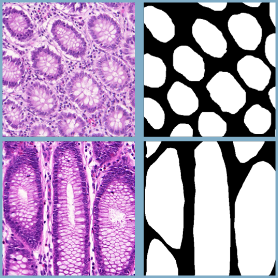
(Kumar et al., 2019)

SegPC



(Gupta et al., 2021)

GlaS



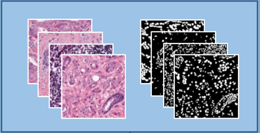
(Sirinukunwattana et al., 2017)

Sparsity is simulated by randomly removing $\rho\%$ of annotations in n_s images.

Experiments

3 baselines

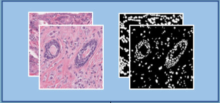
(1) $|\mathcal{D}_s| = 0$



Supervised training

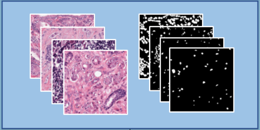
(orignal dataset)

(2) $\mathcal{D}_l$ only



Supervised training

(3) $\mathcal{D}_l \cup \mathcal{D}_s$



Supervised training

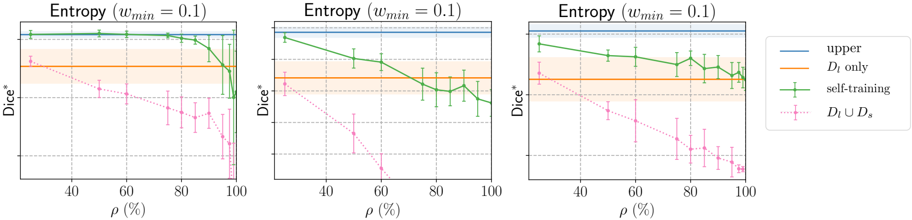
(unlab. pixels assigned to background)

To be considered of interest, our method should be as close as possible to (1) (upper bound) and outperform baselines (2) and (3).

Results

Self-training at fixed n_l

- There is always a cut-off point at which **exploiting additional sparse annotations with self-training becomes beneficial** !
- Self-training struggles at very high data scarcity
- Using $\mathcal{D}_s$ as if it was exhaustively annotated is a bad idea
- For MoNuSeg, the upper baseline is reached with only $\sim$ 30% of the original annotations.



(a) MoNuSeg (b) SegPC (c) GlaS

Results

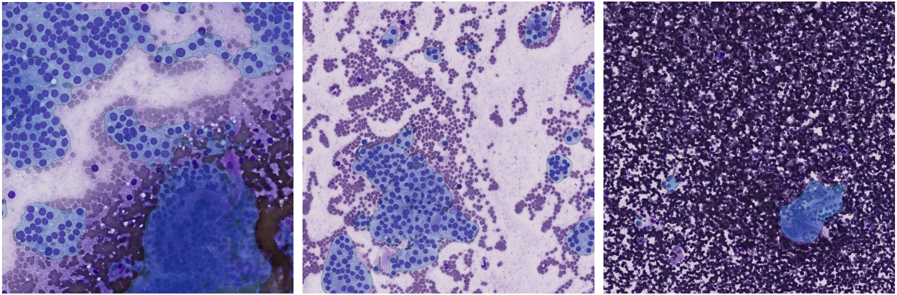
Application to Thyroid FNAB - quantitative

Self-training significantly outperforms the "$\mathcal{D}_l$ only" and "$\mathcal{D}_l \cup \mathcal{D}_s$" baselines.

Method	Dice* (%)
Self-training	89.05 ± 0.85
$\mathcal{D}_l$ only	80.30 ± 5.39
$\mathcal{D}_l \cup \mathcal{D}_s$	83.62 ± 3.52

Results

Application to Thyroid FNAB - qualitative



Conclusion

Self-training can be used to obtain competitive binary segmentation performance with less annotations !

However, let's nuance:

- self-training performance margins (compared to the baselines) are dataset-dependant
- self-training requires a bit of tuning (*i.e.* hyperparameters)

In the future, we plan to:

- further investigate what labeling strategy is more efficient for new datasets
- implement the algorithm in the Cytomine application (batch and interactive)

Thank you !

Pseudo-labels

Generating a pseudo-label $y_{ij}^{(pl)}$ for an unlabeled pixel from the model prediction $\hat{y}_{ij}$ for this pixel:

- Soft label: use $\hat{y}_{ij}$ as-is
- Hard label: 1 if $\hat{y}_{ij} > T$, 0 otherwise. T is an auto-calibrated threshold.

Weighting strategies

We weight the pixel contribution in the loss:

$$\mathcal{L} = \frac{1}{|\mathbf{y}|} \sum_i \sum_j w_{ij} \ell(\hat{y}_{ij}; y_{ij})$$

The weighting strategy is an hyperparameter:

- **Constant**: $w_{ij}^{(cst)} = C > 0$ where C is an hyperparameter
- **Entropy**: $w_{ij}^{(ent)}$ is the entropy of the model prediction $\hat{y}_{ij}$
- **Consistency**: $w_{ij}^{(cty)}$ is a consistency score between model predictions of pixel (i, j) and close pixels
- **Merged**: $w_{ij}^{(mgd)}$ combines the *entropy* and *consistency* strategies

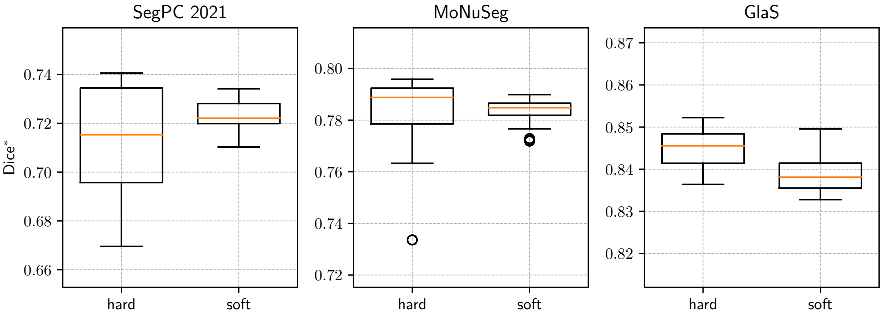
Eventually, w_{ij} is obtained by normalizing the weights computed over a patch so that they sum to 1.

Results

Hard vs. soft labeling

For the given data scarcity regime ($\rho = 90\%$):

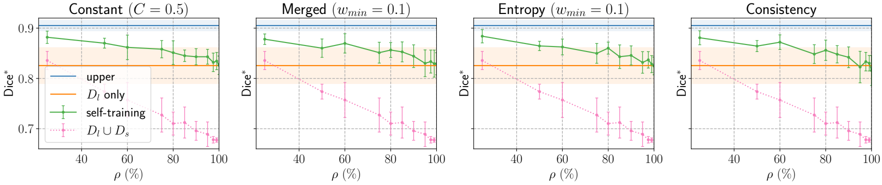
- The best performance are obtained with hard labels.
- Soft labeling yields more stability as performance are less impacted by the choice of a weighting strategy



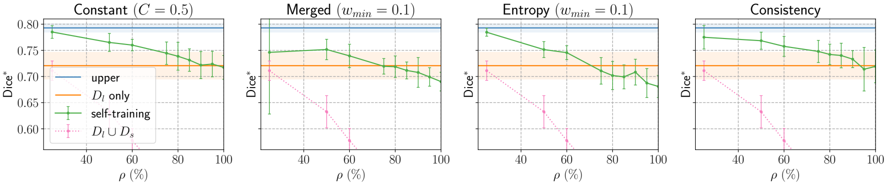
Results

Self-training at fixed n_l - SegPC and GlaS

(a) Glas



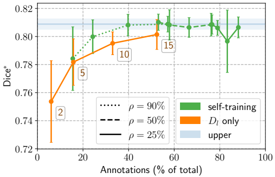
(b) SegPC



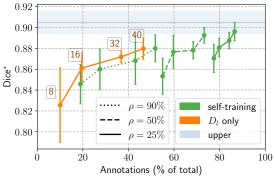
Results

Label sparsely or exhaustively ?

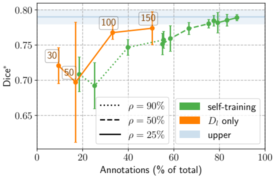
- The answser is dataset-dependant !
- MoNuSeg: annotation budget better spent on sparse labeling (later used with self-training)
- Others: annotation better spent on exhaustive labeling and using supervised training



(a) MoNuSeg



(b) GlaS



(c) SegPC