A task-dependent active learning method for axon segmentation with CNNs

P. Grüning[1*], A. Palumbo[2,3], M.Zille[2,3], E. Barth[1], and A. Madany Mamlouk [1]

[1] Institute for Neuro- and Bioinformatics, University of Lübeck, Lübeck, Germany
[2] Fraunhofer Research Institution for Marine Biotechnology and Cell Technology, Lübeck, Germany
[3] Institute for Experimental and Clinical Pharmacology and Toxicology, University of Lübeck, Lübeck, Germany
[*] Corresponding author, email: gruening@inb.uni-luebeck.de

Abstract: Convolutional neural networks (CNNs) provide reliable segmentation results on biomedical images. However, they can only develop their full potential with a representative dataset. Unfortunately, a large dataset is hard to create in biomedical research, since labeling images is time consuming and requires expert knowledge. Active learning seeks to determine those images that will yield the best results, which effectively reduces labeling cost. We present an active learning method for the stepwise identification of images that should be labeled next and test this method on an axon segmentation dataset. We outperform a baseline and a state-of-the-art method.

I. Introduction

With the use of convolutional neural networks (CNNs), many biomedical computer vision challenges are tackled more and more successfully. However, the quality of the respective CNN largely depends on a large amount of data. This poses a problem, especially for the segmentation of biomedical images, since expert knowledge is often needed and labeling is a time-consuming task. If only a limited amount of time for data labeling is available, only those samples that are most likely to improve the performance should be labeled. We propose a novel, task-driven approach for image segmentation that, given an already trained model and a set of not yet labeled images, will determine which images should be labeled next to improve the generalization performance of the model. We show on an axon segmentation dataset that i.) this strategy outperforms the random picking of images as well as an ensemble-based method and that ii.) a better performance can be reached with fewer labeled images.

This paper focuses on pool-based active learning (AL) [1] in segmentation: a small set of labeled data is already given, while there exists a larger portion of unlabeled data $\mathcal{U}$. A machine learning model can select a certain amount of data from $\mathcal{U}$ as queries for an *oracle*. This oracle can be a human annotator for example, who subsequently labels the queries. $\mathcal{L}$ is updated and the whole process can be repeated. For CNNs, the sample technique *query-by-committee* is often used: an ensemble is trained by using bagging and the variance of the ensemble's output for a sample measures its uncertainty [2].

Many strategies use an influence measure to enforce selected queries to be similar to a large number of samples in $\mathcal{U}$ [2], [3]. Finding the right samples is often transformed into a *set-cover* task [2]. Our strategy incorporates the computation of the *expected error reduction* if a sample is labeled. Konyushkova et al.[4] showed an impressive error-reduction strategy by training a regression model that can predict the expected error reduction of a query, using decision trees rather than CNNs.

II. Material and methods

A dataset $\mathcal{X} = \mathcal{U} \cup \mathcal{L}$ is given, with only a small set $\mathcal{L}$ of labeled data and many unlabeled data in $\mathcal{U}$. In addition, a model $f_{\mathcal{L}}$ is trained on $\mathcal{L}$ and a budget b defines the maximum number of datapoints we can choose as queries. The goal is then to find a subset of $\mathcal{Z} \subseteq \mathcal{X}$ with $|\mathcal{Z} \backslash \mathcal{L}| = b$ and $\mathcal{Z} \cap \mathcal{L} = \mathcal{L}$, that leads to the best outcome. In our particular case, $\mathcal{X}$ would be a set of images $\vec{x_i}$, $i = 1, \dots, N$.

For each image pair $(\vec{x_i}, \vec{x_j})$, our approach aims to numerically answer the following question: *"If we train a network on image $\vec{x_i}$, how well will it predict $\vec{x_j}$?"*. This defines a similarity measure, by assuming that $\vec{x_i}$ and $\vec{x_j}$ are similar if a network trained on $\vec{x_i}$ will predict $\vec{x_j}$ well. To a certain extent, this should also hold true for a noisy or imperfect label. If we have a network $f_{\mathcal{L}}$ already trained on a few images, it is easy to generate a pseudo label $\vec{\hat{y}_i}$ for each image $\vec{x_i}$:

$$\vec{\hat{y}_i} = f_{\mathcal{L}}(\vec{x_i}). \qquad (1)$$

Hence, we define image similarity as:

$$\mathrm{d}(\vec{x_i}, \vec{x_j}) = \mathrm{g}(\mathrm{f}_{\vec{x_i}}(\vec{x_j}), \vec{\hat{y}_j}). \qquad (2)$$

$f_{\vec{x_i}}(\vec{x_j})$ is the prediction of $\vec{x_j}$ by a network $f_{\vec{x_i}}$ trained from scratch on $\vec{x_i}$. g is a quality measure, in our case we use the

Paper ID: 025, DOI: 10.18416/AUTOMED.2020

dice score. Note that we define $d(\overline{x}_\imath, \overline{x}_\imath) = 1$, which is the maximum dice-score.

When picking a subset $\mathcal{Z}$, we have b values for each $\overline{x_\jmath}$ that can be used to model an expected dice score. Hence, we defined the expected performance on an image $\overline{x_\jmath}$ to be:

$$p(\overrightarrow{x_\jmath}) = \frac{1}{b} \sum_{\overrightarrow{x_z} \in \mathcal{Z}} d(\overrightarrow{x_z}, \overrightarrow{x_\jmath}). \qquad (3)$$

To avoid exhaustively training a network for each image $\overrightarrow{x_\imath} \in \mathcal{X}$, we picked a random image set $\overrightarrow{x_k} \in \mathcal{K} \subseteq \mathcal{U}$, using a heuristic to approximate the image similarities. Both similarities between two points $\overrightarrow{x_\imath}$ and $\overrightarrow{x_\jmath}$ are set to the known value, if only one is known. For unknown values, we look at all paths that start from image $\overrightarrow{x_\imath}$, move to $\overrightarrow{x_k}$ and then go to $\overrightarrow{x_\jmath}$, defining the final similarity as follows:

$$d(\overrightarrow{x_\imath}, \overrightarrow{x_\jmath}) = \max_{\overrightarrow{x_k} \in \mathcal{K}} \frac{1}{2}\left(d(\overrightarrow{x_\imath}, \overrightarrow{x_k}) + d(\overrightarrow{x_k}, \overrightarrow{x_\jmath})\right). \qquad (4)$$

In conclusion, the algorithm is defined as:

1. Compute pseudo labels $\overrightarrow{\hat{y}_\imath}$ with model $f_{\mathcal{L}}$ for all $\overrightarrow{x_\imath} \in \mathcal{X}$.
2. Pick a random subset of images $\overrightarrow{x_k} \in \mathcal{K} \subseteq \mathcal{U}$.
3. For each $\overrightarrow{x_k}$, train a network $f_{\overrightarrow{x_k}}$ from scratch.
4. Compute the similarities $d(\overrightarrow{x_k}, \overrightarrow{x_\imath})$, $\overrightarrow{x_k} \in \mathcal{K}, \overrightarrow{x_\imath} \in \mathcal{X}$.
5. Compute the remaining similarities $d(\overrightarrow{x_\imath}, \overrightarrow{x_\jmath})$.
6. Find the subset $\mathcal{Z}$ that maximizes $\min_{\overrightarrow{x_\jmath} \in \mathcal{X}} p(\overrightarrow{x_\jmath})$.
7. Update $\mathcal{L} \coloneqq \mathcal{Z}$.

III. Data & experiments

We created a dataset for axon segmentation of murine primary cortical neurons. A microfluidic device is used to separate neuronal cell bodies from axons and grayscale images of the axons are captured via phase contrast microscopy. We determined a training and test set containing 16 and 26 images respectively. Labeling one image takes approximately one hour. For our CNN, we used a classical u-net architecture [5], slightly modified with batch normalization. For each dataset, we did 5 *experiments*. Each experiment started with 3 randomly drawn labeled images. We trained on subsets with increasing size $B = \{5, 7, 12, 15, 16\}$. On each subset, a neural network was trained and evaluated on the test set.

As a *baseline*, we picked the images randomly. We did this eight times and averaged the test scores to estimate the expected outcome. Additionally, we compared our approach with an ensemble based method [2]. At each step, we trained an ensemble of 4 networks. 3 were trained on a random subset containing roughly 80% of the current training data (bagging) and a fourth network was trained on the full dataset. We used the mean KL-Divergence between each network's output and the mean output of all networks to measure uncertainty. We selected half of the images from $\mathcal{U}$ with the highest uncertainty as our *candidates*. The network trained on the full dataset was used to extract feature vectors from each image in $\mathcal{U}$, by global average pooling of the last feature layer. This method defined similarity of two images as the cosine similarity of the two feature vectors.

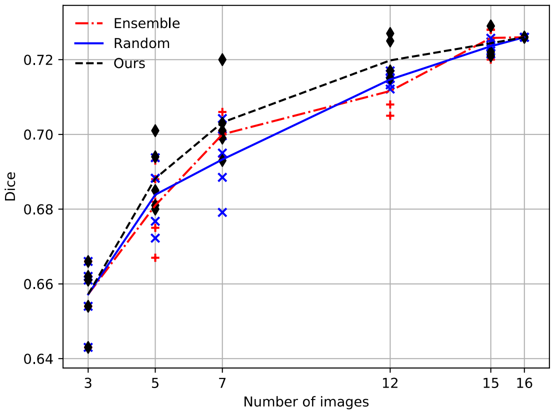

Figure 1: Dice coefficients obtained from the axon data. The size of the training set is plotted against the resulting dice score on the test set. The dashed black line shows the average for our approach, diamonds show the five individual scores. The blue line shows the average random scores, blue crosses the individual scores. Finally, the red dash-dotted line shows average performance of the ensemble approach, red plus signs indicate the scores.

We found the subset of candidates that maximizes the similarity again by exhaustive search in contrast to the greedy algorithm in the original paper. We here refer to this method as the *ensemble* method.

IV. Results & discussion

The results in Figure 1 show that our method outperforms the baseline as well as the ensemble approach on almost all image subsets. However, in a next step a rigid cross-validation should be done to cover all possible test runs in our experiments and to better understand the intricacies of active learning. For example, further research is needed to evaluate what happens if the labeled images differ from the majority of images in the unlabeled set.

Yet, given the proposed setup, our method obtained better intermediate results. Thus, in circumstances where only a specific time budget for labeling is given, or early decisions need to be made based on few labeled data, using active learning to pick the right samples might help to reach desired performance levels while saving valuable man-hours spent on labelling resources.

Author's statement

The author state no funding involved and state no conflict of interest.

References

[1] B. Settles, "Active Learning Literature Survey," 2009.
[2] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in International conference on medical image computing and computer-assisted intervention, 2017, pp. 399–407.
[3] S. Dutt Jain and K. Grauman, "Active image segmentation propagation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2864–2873.
[4] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from data," in Advances in Neural Information Processing Systems, 2017, pp. 4225–4235.
[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention, 2015, pp. 234–241.