Supplementary Material for: Every Annotation Counts: Multi-label Deep Supervision for Medical Image Segmentation

Simon Reiß[1,2] Constantin Seibold[1] Alexander Freytag[2] Erik Rodner[2,3] Rainer Stiefelhagen[1]
[1]Karlsruhe Institute of Technology [2]Carl Zeiss AG [3]University of Applied Sciences Berlin
{simon.reiss,constantin.seibold,rainer.stiefelhagen}@kit.edu

Abstract

This document contains supplementary information for the paper Every Annotation Counts: Multi-label Deep Supervision for Medical Image Segmentation. *We provide additional details about implementation and code sources and present several ablation studies for the baselines that we benchmark against in the main paper.*

1. Implementation Details

In the main paper, we described in detail which hyperparameters and training setup we used to train UNets [8] for retinal fluid segmentation. More precisely, we used the PyTorch framework [5] for implementation. Our UNets all expand the implementation as found in [1], where we use the bilinar interpolation variant as opposed to the transposed convolutional up-scaling in the decoder. For models using the invariant information clustering (IIC) loss [3], we reused code from the author's official implementation [2] (i.e. the loss and geometric transformation snippets).

UNet decoder feature maps. In the main paper and the following ablations, we refer to feature maps of the UNet decoder as $f_0, \ldots, f_4$. Referring to the implementation in [1], we can directly outline which feature maps we use:

$$f_0 : x_5 \tag{1}$$
$$f_1 : \text{up1}(x5, x4) \tag{2}$$
$$f_2 : \text{up2}(x, x3) \tag{3}$$
$$f_3 : \text{up3}(x, x2) \tag{4}$$
$$f4 : \text{up4}(x, x1) \tag{5}$$

The definitions of what these variables refer to are found in the forward pass in *unet_model.py*.

2. Baseline Ablation Studies

To make sure we have strong baselines to benchmark our approach against, we carried out ablation studies for both the *IIC Baseline* and *MIL Baseline* on the validation set and investigated the effect of using different decoder layers for deep supervision.

#epochs	$\mathcal{U}$	#clusters	f	validation (mIoU)
100		–	–	62.42 ± 4.11
200		–	–	62.54 ± 3.88
100	✓	5	f_4	63.42 ± 4.32
100	✓	10	f_4	64.33 ± 2.84
100	✓	20	f_4	64.63 ± 3.36
100*	✓	10	$f_{\{0-4\}}$	$\mathbf{65.23 \pm 3.58}$

Table 1. Ablation for the number of heads for IIC models. Due to high memory consumption, we reduce the batch size from 16 to 8 for the model with the * symbol. All models are trained with full access to pixel-wise annotated masks.

IIC baseline clusters. In Table 1, we ablate the hyperparameter for IIC-based models of how many clusters/output-heads should be used. We evaluate this in the full access setting and, as described in the main paper, leverage a standard pixel-wise cross-entropy loss on the outer most feature map f_4 alongside the IIC loss for all images. We see that the IIC loss adds to the segmentation accuracy even when all pixel-wise labels are present. As IIC requires two forward passes with differently perturbed images, we might suspect that the additional iterations are the reason for the accuracy increase. Therefore, in the first and second row, we show a standard UNet trained for as many and twice as many epochs as the IIC models to rule out this suspicion. The results indicate only a marginal alteration in mean Intersection over Union (mIoU).

The best accuracy for IIC models is achieved using 20 output-heads. However, the relatively small accuracy gain as compared to 10 heads does not justify the increased memory consumption and training time. Thus, the IIC models in the paper always use 10 output-heads. Integrating the IIC loss deep into network feature maps (10 output-heads for layers $f_{\{0-4\}}$) led to the overall best configuration, even though batch size had to be lowered from 16 to 8 due to

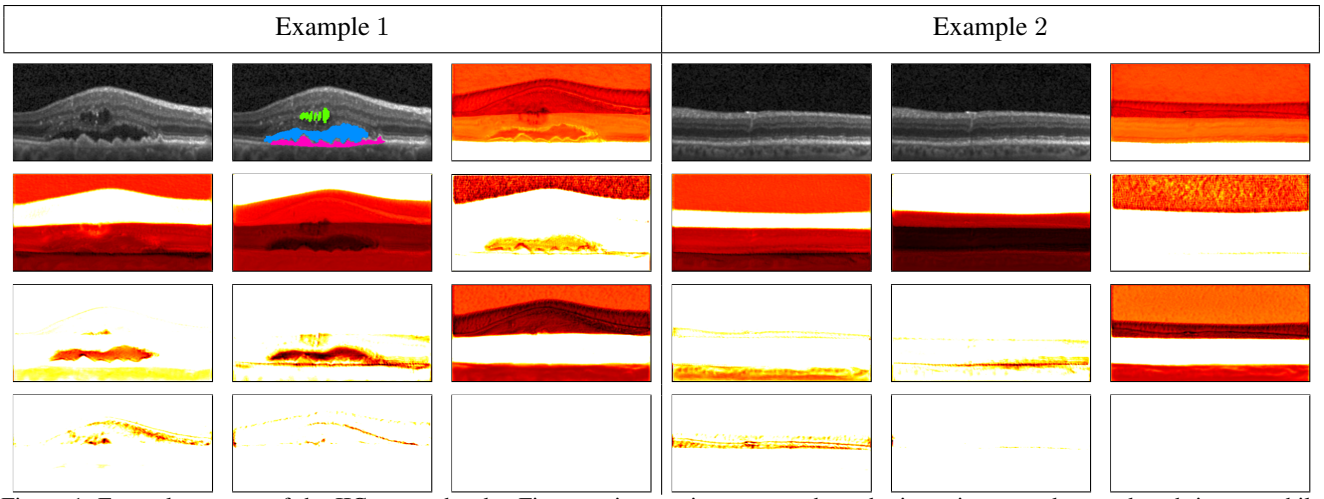

Figure 1. Example outputs of the IIC output-heads. First two images in row one show the input image and ground-truth image, while remaining images show feature scaled outputs of the 10 IIC heads.

the added memory requirement. We did a qualitative investigation of the neighborhood displacement hyperparameter, i.e. the padding d in the implementation (see [3] *Section 3.3 Implementation* for further details) and opted for $d = 5$.

In order to show what the output-heads actually learn during their unsupervised training, we show two examples in Fig. 1. The visualization is done by feature scaling of the outputs for each of the 10 heads. Example 1 shows the output for a diseased optical coherence tomography (OCT) b-scan while example 2 shows the output when a healthy b-scan serves as input. We observe that many of the IIC output-heads capture different physical layers of the retina (which is a task of its own for retinal image analysis). That this leads to an improvement in a model's accuracy is therefore quite apparent, as some methods use pre-processing techniques to explicitly derive information from retina layers [4].

Deep supervision integration. In Table 2, we investigated in which layers of the UNet's decoder deep supervision should be introduced, if any. For this analysis, we compare the standard UNet accuracy when using 24 pixel-wise annotated masks (first row) with Multiple Instance Learning (MIL)-based models adding supervision from image-level labels. The results indicate nicely that adding deep supervision in all decoder layers yields the highest mIoU on the validation set. As such, our *Deeply Supervised MIL* and *Deeply Supervised IIC* models enforce their respective loss functions on feature maps $f_{\{0-4\}}$.

MIL baseline pooling function. As a lower bound for the semi-weakly supervised scenario in the main paper, we leverage the MIL baseline model. Closest to what we do is [6], although we use a binary cross-entropy loss to utilize the image-level labels and of course have additional access to few pixel-wise masks. In Table 3, we compared a

$\mathcal{G}$	f	validation (mIoU)
	–	46.84 ± 6.49
✓	f_4	49.48 ± 4.88
✓	$f_{\{3,4\}}$	50.13 ± 6.25
✓	$f_{\{2,3,4\}}$	51.81 ± 5.58
✓	$f_{\{1,2,3,4\}}$	51.47 ± 4.03
✓	$f_{\{0,1,2,3,4\}}$	$\mathbf{53.52 \pm 4.69}$
✓	$f_{\{0,1,2,3\}}$	52.85 ± 4.50
✓	$f_{\{0,1,2\}}$	51.81 ± 6.25
✓	$f_{\{0,1\}}$	51.68 ± 5.48
✓	f_0	50.23 ± 7.38

Table 2. Ablation for enforcing the image-level MIL loss on different feature maps within the UNet. First line indicates performance without image-level labels, second line indicates what we refer to in the paper as *Baseline MIL*, the best performing model here is the *Deeply Supervised MIL* model in the paper.

$\mathcal{G}$	pooling function	validation (mIoU)
	–	46.84 ± 6.49
✓	max pooling	46.96 ± 12.73
✓	average pooling	$\mathbf{53.52 \pm 4.69}$

Table 3. Ablation of pooling functions for MIL-based models with access to 24 pixel-wise masks and global labels are provided.

standard UNet (first row) with two *Deeply Supervised MIL* models in a setting using only 24 pixel-wise masks. We can observe that the choice of the pooling function for aggregating feature maps into image-level predictions is important: max pooling did not significantly improve the segmentation accuracy, while average pooling did. Therefore, all our *MIL Baseline* models employ average pooling to aggregate features and introduce image-level semantics.

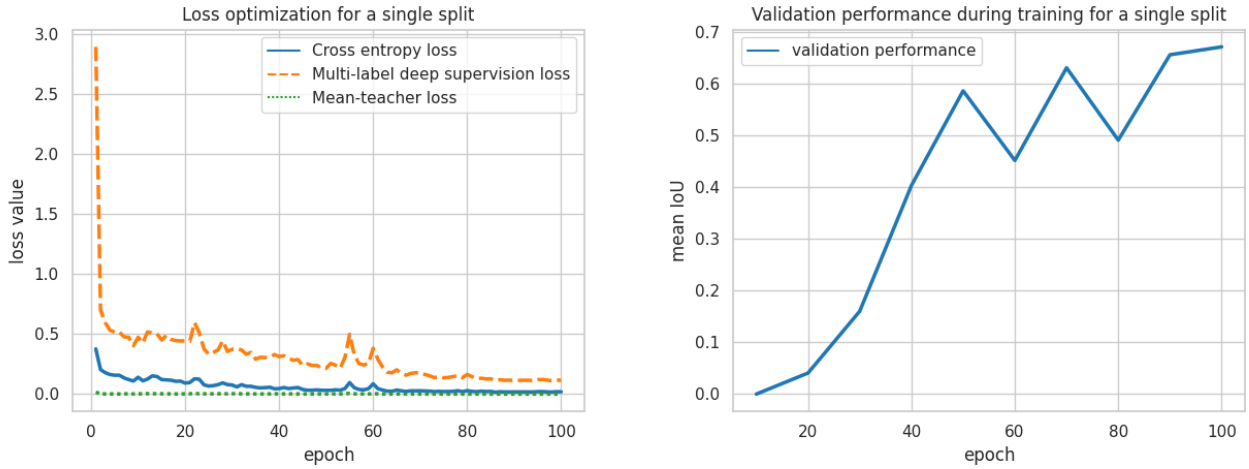

Figure 2. Exemplary loss decay and validation performance for training our best method *Mean-Taught Deep Supervision* with 24 pixel-wise masks. Left: optimization process for a single split, displaying cross-entropy loss, multi-label deep supervision loss and the mean-teacher regularization (mean-squared error) loss, right: corresponding validation performance in mean IoU for the same split every 10 epochs.

Method	$\mathcal{G}$	$\mathcal{U}$	validation $\mathcal{M}(24)$	testing $\mathcal{M}(24)$
Baseline [8]			46.84 ± 6.49	48.63 ± 5.17
Multi-Label Deep Supervision			52.03 ± 5.48	52.68 ± 6.82
IIC Baseline[8] [3]		✓	52.82 ± 7.17	53.08 ± 6.13
Deeply Supervised IIC[8]		✓	53.63 ± 4.69	50.10 ± 7.92
Perone et al.[10] [7]		✓	56.51 ± 5.56	54.75 ± 5.96
Self-Taught Deep Supervision		✓	54.13 ± 7.38	56.11 ± 6.30
Mean-Taught Deep Supervision[10]		✓	60.63 ± 5.35	58.84 ± 6.57
MIL Baseline	✓		49.48 ± 4.88	49.07 ± 8.20
Deeply Supervised MIL	✓		53.52 ± 4.69	51.13 ± 3.93
Self-Taught Deep Supervision	✓		57.80 ± 4.68	59.29 ± 7.52
Mean-Taught Deep Supervision[10]	✓		61.36 ± 4.73	60.45 ± 5.71

Table 4. Validation- and testing accuracy (mIoU) when training different methods using 24 pixel-wise annotations; $\mathcal{G}$: additional usage of global labels, $\mathcal{U}$: additional usage of unlabeled images. Superscript in method names indicate a batch size different than 16.

3. Training progress in low-data scenarios

As suggested in the reviews, we visualize the training progress of the *Mean-Taught Deep Supervision* model in Fig. 2. For the model trained with 24 pixel-wise labeled masks and global labels, the progression of training loss (left figure) and validation accuracy (right figure) over the 100 training epochs are shown. The figure underlines that the original mean-teacher regularization has a rather small contribution to the overall loss as compared to our multi-label deep supervision loss. As described in the main paper, we applied early stopping for all training runs. In consequence, we use the model with the best validation accuracy to evaluate the testing images. To show the effects when moving from validation data to test data, we include Table 4 which contains average validation and average testing accuracy across the 10 splits (24 pixel-wise mask scenario). The *Mean-Taught Deep Supervision* method achieves a validation accuracy of 61.36 ± 4.73 with testing accuracy amounting to 60.45 ± 5.71 mean IoU. We do not see statistical significant differences between results on validation and test set.

References

[1] Milesi Alexandre. Pytorch-unet. https://github.com/milesial/Pytorch-UNet/tree/master/unet. Accessed: 2020-11-17. 1

[2] Xu Ji. Invariant information clustering for unsupervised image classification and segmentation. https://github.com/xu-ji/IIC. Accessed: 2020-11-17. 1

[3] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019. 1, 2, 3

[4] Donghuan Lu, Morgan Heisler, Sieun Lee, Gavin Ding, Marinko V Sarunic, and Mirza Faisal Beg. Retinal fluid segmentation and detection in optical coherence tomography im-

ages using fully convolutional neural network. *arXiv preprint arXiv:1710.04778*, 2017. 2

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith 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. 1

[6] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014. 2

[7] Christian S Perone and Julien Cohen-Adad. Deep semi-supervised segmentation with weight-averaged consistency targets. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 12–19. Springer, 2018. 3

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 3