Boosting Liver and Lesion Segmentation from CT Scans by Mask Mining

Karsten Roth[1,2], **Jürgen Hesser**[1,2], **Tomasz Konopczyński**[1,2,3]
[1]Department of Radiation Oncology, University Medical Center Mannheim,
[2]IWR, ZITI, Heidelberg University,
[3]Tooploox
karsten.roth@stud.uni-heidelberg.de
juergen.hesser@medma.uni-heidelberg.de
tomasz.konopczynski@tooploox.com

Abstract

We propose a novel procedure to improve liver and lesion segmentation from CT scans for U-Net based models. Our method extends standard segmentation pipelines to focus on higher target recall or reduction of noisy false-positive predictions, by including segmentation errors into a new learning process appended to the main training setup. This allows the model to find features which explain away previous errors. We evaluate on semantically distinct architectures on liver and lesion segmentation data are provided by the Liver Tumor Segmentation challenge (LiTS), with an increase in dice score of up to 2 points.

1 Introduction

Liver imaging nowadays is mostly done via Computed Tomography (CT)[1]. Providing fully-automatic segmentation of liver and lesion tissue from CT data can hence be a useful tool to help with diagnosis and treatment planning. Common approaches utilize U-Nets[10], e.g. [8, 1, 4]. However, training of neural networks can be a difficult endeavour. To improve on existing scores, computationally expensive re-runs without guarantee of improvement are often needed.

We thus propose a novel pipeline to reliably boost network segmentation performances by including segmentation errors as novel training masks in a post-training step. Using segmentation error types and specific loss functions as new training signals, we are able to offer a framework helping networks explain away own segmentation errors, thereby boosting segmentation performance. This also means that our method stays independent of architecture and data choices, and allows for improved performance without costly retraining. Unlike approaches such as [12, 11], who propose a Tversky-coefficient based loss adding additional hyperparameters to penalize false-positive or false-negative predictions during training, or [9] who utilize segmentation error types in an adversarial setup to train refinement networks. Where [12] introduces two new hyperparameters and [9] train a complex adversarial setup limiting usable network complexities, both closely link the inclusion of errors to the learning setup. This requires heavy tuning for different architectural setups, especially going to three-dimensional data, which is common for many medical segmentation tasks.

To examine the architecture-independent applicability of our method, we test on distinctly different architectures, focused around 2D and 3D U-Net [10, 3, 13] pipelines, trained and evaluated on the Liver Tumor Segmentation (LiTS) dataset [1].

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

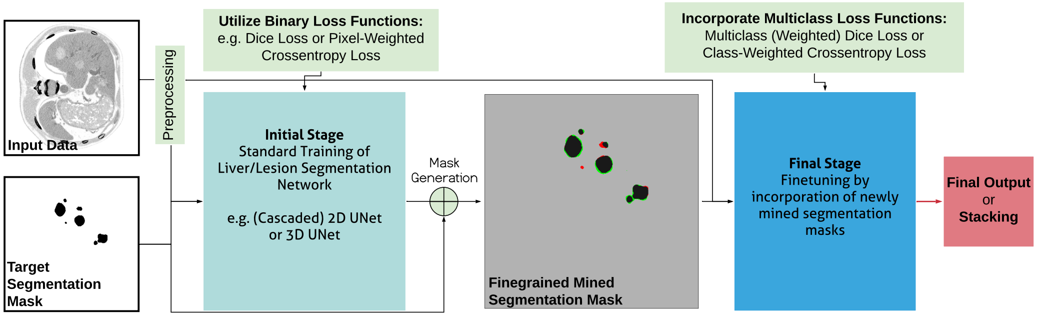
Input Data
Target Segmentation Mask
Preprocessing
Utilize Binary Loss Functions:
e.g. Dice Loss or Pixel-Weighted Crossentropy Loss
Initial Stage
Standard Training of Liver/Lesion Segmentation Network
e.g. (Cascaded) 2D UNet or 3D UNet
Mask Generation
Finegrained Mined Segmentation Mask
Incorporate Multiclass Loss Functions:
Multiclass (Weighted) Dice Loss or Class-Weighted Crossentropy Loss
Final Stage
Finetuning by incorporation of newly mined segmentation masks
Final Output
or
Stacking

Figure 1: *The Mask Mining Pipeline.* Learned segmentation masks are compared with the ground truth masks to generate new finegrained training masks containing previously made segmentation errors. Retraining on those allows the network to learn to explain away mistakes.

2 Inclusion of Segmentation Errors through Mined Masks

Fundamental for our proposed pipeline extension (fig. 1) is the generation of new training masks to alter the current network performance and allow the network to learn from its own errors.
Basic Setup: A segmentation pipeline of choice is trained until convergence following any training procedure. Now, segmentation masks for the training data are generated through single forward passes with minimal computational burden. These masks are then compared to the original ground truth to determine new training classes for each pixel/voxel, based on segmentation error cases: *True Negative*, *False Positive*, *False Negative* and *True Positive*. This gives four target classes compared to the binary case with two classes. We then append four single-layer output channels serving as error prediction layers to the output layer. This introduces no relevant new parameters, but ensures that all previously learned weights are kept until finetuning. Finally, retraining on the novel masks is performed. Due to the initial pretraining, *convergence occurs much faster*.
Relevance of loss function: The choice of loss separates the retraining into two equivalent setups. Commonly after training, the majority of predicted pixels are true positive or negative.
A pixel-weighted crossentropy loss (pwce) (e.g. [10]) gives highest learning signal to high frequency targets. As we have a high imbalance towards true positive/negative predictions, retraining on error masks primarily reinforces these predictions while dropping noisy false positives. The retrained, now multiclass, error case predictions are then grouped into true positive/false negative predictions and true negative/false positive predictions to generate the final new binary segmentation mask.
A dice-coefficient based loss (e.g. [4]) injects a stronger learning signal for underrepresented classes for higher recovery of false-negative/positive pixels. Here, the primary interest lies in explaining away obfuscating features while retaining crucial ones, so the true positive error mask class is replaced with the ground truth segmentation mask. This allows the network to transfer properties generating false-positives to the respective output channel and recover generators for false-negative predictions. The final segmentation is taken directly from the true positive output channel.
Both losses offer a performance boost and are mentioned for completeness. *However, for all subsequent results a dice-based loss is utilized, as it provides marginally higher improvements.*

3 Application to Liver and Lesion Segmentation and Conclusion

Network Architecture: We investigate the performance of our method on liver and lesion segmentation by evaluating dice score performance on distinct architectures: *(i) Cascaded 2D*[2], which trains a 2D segmentation network for liver and lesion segmentation separately, *(ii) Cascaded 3D*, which does the same for a 3D setup and *(iii)*, *Combined Cascaded 2D*[13], which trains separate segmentators for liver and lesion in a simultaneous setup. All networks use common extensions such as multislice inputs[5], batch normalization[7] or residual blocks[6].Each pipeline is trained to convergence before applying our extension to ensure that we do not just prolong the training process. Initial training is done using a pwce loss with distance-transformation weightmaps (see [10]) for liver and a loss based on dividing the pwce loss, L^{pwce}, by a smooth dice score L^{dice} (see e.g. [4]).
LiTS dataset: The Liver Tumor Segmentation (LiTS) dataset[1] contains 131 3D CT scans of the lower abdominal area with ground truth masks for liver and lesion tissue, as well as 70 test volumes, which are evaluated by online submission to the dataset webpage. All volumes have horizontal

Table 1: *Quantitative evaluation of network performance before and after error inclusion (Inc.).* We show volume-averaged dice scores for liver and lesion segmentation on the test set and fixed training and validation sets. We see a clear improvements in dice scores. In addition, error inclusion reduces seed-dependent variation in performance (measured over three runs).

Setup	Training Dice		Validation Dice		Online Test Dice	
	Liver	Lesion	Liver	Lesion	Liver	Lesion
2D	96.9 ± 0.3	71.9 ± 0.4	95.9 ± 0.3	63.5 ± 0.6	95.3 ± 0.2	62.9 ± 0.3
Inc.	**97.0 ± 0.1**	**73.7 ± 0.2**	**96.3 ± 0.2**	**64.9 ± 0.2**	**95.5 ± 0.3**	**63.5 ± 0.2**
3D	92.2 ± 1.4	63.0 ± 0.8	91.4 ± 0.9	56.8 ± 2.0	91.2 ± 1.0	55.5 ± 0.9
Inc.	**94.2 ± 0.3**	**66.1 ± 0.4**	**91.8 ± 0.6**	**57.7 ± 0.4**	**92.0 ± 0.4**	**56.5 ± 0.2**
Cmb	94.5 ± 0.3	70.1 ± 0.5	92.9 ± 0.7	61.6 ± 0.5	93.4 ± 0.3	61.9 ± 0.2
Inc.	**96.2 ± 0.5**	**72.3 ± 0.4**	**94.0 ± 0.3**	**63.4 ± 0.4**	**94.7 ± 0.3**	**63.0 ± 0.1**

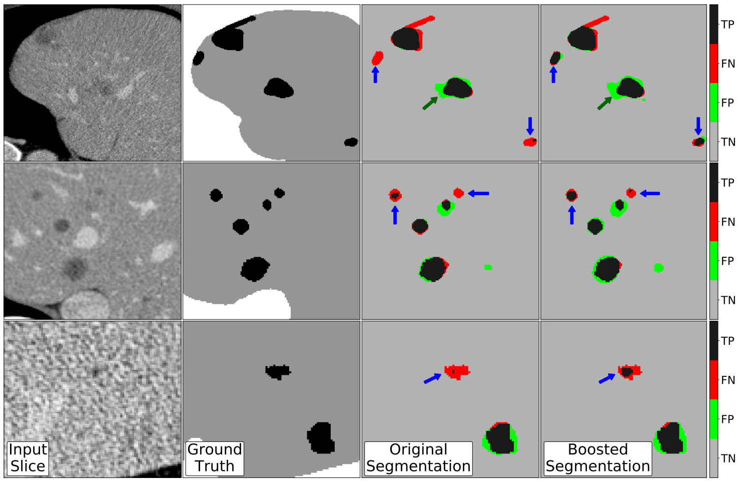

Figure 2: *Qualitative evaluation before/after error mask retraining. Original Segmentation* denotes the segmentation errors for the initial network, while *Boosted Segmentation* shows changes after retraining: The number of false negatives (*red*) is clearly reduced, with a slight introduction of new additional false positives (*green*). *Grey* and *black* denote true negative and positive errors.

dimensions of 512 with near constant resolution. In the axial direction dimensionality and resolution vary strongly, which is a relevant factor for any approach using higher-than-two dimensional data input. Before training, the data is bounded to $[-100, 600]$ HU before performing normalization. For evaluation, only the largest connected component is used to generate the final liver segmentation.

Results: We compute the averaged dice score per volume on the test volumes before and after application of our method for all architectures. Here, relative improvement is the key metric to examine. Results are summarized in tab. 1, showing a consistent gain over the initially trained model, especially for the combined training setup. This is arguably due to the simultaneous boost in liver and lesion segmentation performance.

The inclusions of mined trained masks into the training process specifically benefits validation performance. This is rooted in the splitting procedure, as training and validation set are drawn from the same sample set. Due to different sources contributing to the dataset [1], the test set samples therefore differ much stronger from the training set. Newly mined features are therefore more expressive on the validation set.

Conclusion: We introduced a novel extension to standard liver and lesion segmentation pipelines on the basis of the Liver Tumor Segmentation (LiTS) dataset. By helping the network learn and thereby explain away previously made errors using automatically generated training labels, we boost segmentation performance on different and distinct architectures and training styles. Due to the architecture-independent applicability we are certain that our method can be extend to other medical image segmentation problems.

References

[1] Patrick Bilic et al. The liver tumor segmentation benchmark (lits). *CoRR*, abs/1901.04056, 2019.

[2] Patrick Ferdinand Christ et al. Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3d conditional random fields. *CoRR*, abs/1610.02177, 2016.

[3] Özgün Çiçek et al. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016.

[4] Michal Drozdzal et al. The importance of skip connections in biomedical image segmentation. *CoRR*, abs/1608.04117, 2016.

[5] Xiao Han. Automatic liver lesion segmentation using A deep convolutional neural network method. *CoRR*, abs/1704.07239, 2017.

[6] K. He et al. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[8] Fabian Isensee et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *CoRR*, abs/1809.10486, 2018.

[9] Mina Rezaei et al. Conditional generative refinement adversarial networks for unbalanced medical image semantic segmentation. *CoRR*, abs/1810.03871, 2018.

[10] Olaf Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[11] Karsten Roth, Tomasz K. Konopczynski, and Jürgen Hesser. Liver lesion segmentation with slice-wise 2d tiramisu and tversky loss function. *CoRR*, abs/1905.03639, 2019. URL `http://arxiv.org/abs/1905.03639`.

[12] Seyed Sadegh Mohseni Salehi et al. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *Machine Learning in Medical Imaging*, pages 379–387, 2017.

[13] E. Vorontsov et al. Liver lesion segmentation informed by joint liver segmentation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1332–1335, April 2018.