

# Triple-View Feature Learning for Medical Image Segmentation

Anonymous

Anonymous

**Abstract.** Deep learning models, especially supervised Encoder-Decoder style networks, exhibit promising performance in medical image segmentation, but come with a high labeling cost. We propose a co-training semantic segmentation framework, TriSegNet, which is semi-supervised. It uses *triple-view* feature learning on a limited amount of labeled data and a large amount of unlabeled data. The triple-view architecture consists of three different pixel-level classifiers and a low-level shared-weight learning module. The model is first initialized with labeled data. Label processing, including data perturbation, confidence label voting and unconfident label detection for annotation, enables to train model on labeled and unlabeled data simultaneously. The low-confidence model gets improved via other two-view feature learning with an additional stage. This process is repeated until each model reaches the same confidence level as its counterparts. The strategy is designed to enable triple-view learning of generic medical image datasets. Bespoke overlap-based and boundary-based loss functions are tailored to the different stages of training. The segmentation results are evaluated on four publicly available benchmark datasets including Ultrasound, CT, MRI, and Histology images. Repeated experiments demonstrate the effectiveness of the proposed network compared against other semi-supervised algorithms, across a large set of evaluation measures.

## 1 Introduction

The promising performance of deep learning for medical imaging relies not only on network architecture engineering, but also on the availability of sufficient high-quality manually annotated data, which is hard to come by. Co-training, and self-training are two widely studied approaches in semi-supervised learning.

**Self-training** first initializes a model with labeled data. Then the model generates pseudo masks for unlabelled data. A condition is set for the selection of pseudo masks, and the model is retrained by expanding its training data [26].

**Co-training** is normally used to train two separate models with two views. They benefit each other [3] by expanding the size of the training data. Deep co-training was first proposed by [17] pointing out the challenge of ‘collapsed neural networks’: training two models on the same dataset cannot enable multi-view feature learning because the two models will necessarily end up similar. Output smearing, diversity augmentation, and dropout for pseudo label editing were designed to mitigate that effect [5].

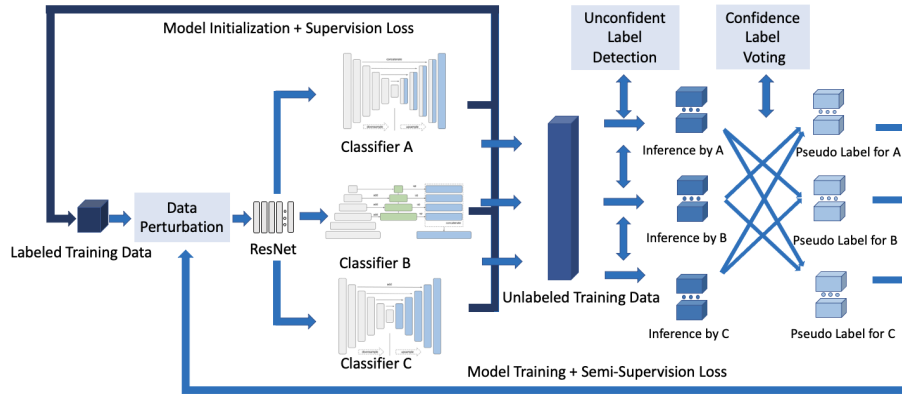


Fig. 1. The Architecture of TriSegNet Framework

## 2 Methodology

The central idea in our method is that several (three in our case) different views of the data and their associated learned parameters are developed in separate models simultaneously. This gives each model a chance to complement the others in what they contribute to the learning process. Moreover, they do not each process the same dataset: rather, they take carefully crafted disjoint parts of the data and learn what they can from each of those subsets.

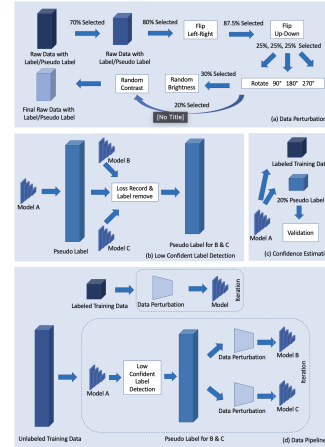
A multi-view co-training semi-supervised learning method for classification is also proposed in [5], but for a classification task, not for semantic segmentation. Detailed mask prediction requires an entirely novel technique. Our pseudo-labels are generated afresh at intermediate steps selecting from amongst the output of two individual CNN components in order to train a third one. Our confidence estimation is different from the uncertainty-aware scheme [23]: the triple framework decides which pseudo-labels to propagate further through selecting those in which the individual component models have had high confidence and have ‘voted’ for. This also has the effect of increasing the overall confidence of the framework in a way that we specifically show how to measure. Pseudo-labels with too low a confidence level are not used in the following step. How low that level of confidence is depends on the stage of the training: the further into the training process, the more confident the framework needs to become. Thus the number of images available for training gets increased gradually.

The architecture of our **Triple-view** feature learning for medical image semantic **Segmentation Network** (TriSegNet) is illustrated in Figure 1. The label editing including unconfident label detection and confidence label voting based on confidence estimation is proposed to improve the feature expression of un-annotated data. Each of the three views of feature learning makes use of random data perturbation for regularization.

A pre-trained ResNet is utilized as a low-level feature learning module, shared between the three high-level feature learning pixel-level classifiers A, B and C. For the purpose of triple-view learning [8], the architecture and parameters initialization are developed separately. A, B, C are Encoder-Decoder-based. In order to transfer sufficient semantic information, they fully model long-range dependencies, variation size of feature expression with low computational cost, model skip connection [18], they bypass spatial information [4], and process multi-scale feature maps [14]. Inspired by [9], the low-level feature learning module is shared, and the three views of classifiers enable the ResNet to extract low-level features in a generic manner. A, B and C not only extract features, but also vote, generate pseudo labels, generally benefiting each other in the semi-supervised process.

## 2.1 Training Setup

We denote by  $\mathcal{L}, \mathcal{U}, \mathcal{T}$  a small labeled dataset, a large size of unlabeled dataset, and a dataset for testing. Each batch of labeled data is  $(x, y_{gt}) \in \mathcal{L}, (x, y_{gt}) \in \mathcal{T}$ ; batches of only raw data are  $(x) \in \mathcal{U}; y_{pred}, y_{pseudo}$  is the dense map generated by different views of TriSegNet for prediction. We split the training process into three stages with an inference stage. **Stage 1** initializes the three views of classifiers  $f_n, n = 1, 2, 3$  with labeled dataset  $\mathcal{L}$ , repeatedly as shown in Fig. 2(d), with data perturbation as per Fig. 2(a). To properly initialize three classifiers with unbalanced data, overlap loss is used. The confidence weight during the training process is estimated and recorded as  $\alpha_n, n = 1, 2, 3$  until inference is complete. Each model estimates its confidence on the labeled data by on randomly selected pseudo labels, also compared against the other two models as sketched in Figure 2(c). **Stage 2** learns from the generated pseudo labels in an iterative manner. Unlabeled data  $(x) \in \mathcal{U}$  are used by  $f_n$  as  $(x, y_{n\_pred})$  pairs. Any data labelled with low confidence is removed. A fresh vote from another model  $f_n$  generates pseudo label  $(x, y_{n\_pseudo}) = \alpha_{n-1} \times f_{n-1}(x) + \alpha_{n+1} \times f_{n+1}(x)$  to train the rest of the model. The loss function at this stage is mixed overlap- and boundary-based loss. **Stage 3** is to train additionally the lowest confident classifier with  $\mathcal{L}, \mathcal{U}$ . The learning rate for all training stage is set as  $2 \times 10^{-4}$ , with an Adam optimizer. The training epoch is set to 150 in Stage 1, 100 in Stage 2, and 150 or early stopped until the lowest confident classifier reaches same confident with other classifiers in Stage 3. The batch size is set to 16. **Inference** is to predict jointly on an unseen image  $(x) \in \mathcal{T}$  to  $y_{pred} = \sum_{n=1}^3 \alpha_n \times f_n(x)$ .



**Fig. 2.** Data Pipeline and Label Processing stages

## 2.2 Label Processing

To encourage meaningful differences among views for co-training, it is essential to pin down each classifier’s architecture, as well as its label editing. In TriSegNet we propose three label editing approaches in the whole training stages: Data perturbation, low confidence label detection, and Confidence label voting.

**Data Perturbation.** Mean Teacher [19] uses perturbation to assess the consistency of the same image under different disturbances. This is extended in Dual Student to pixel-wise tasks in guided collaborative training [13]. Naive-student [6] uses perturbation for pseudo labels in an iterative manner to train the student network. To encourage differences between the three views, TriSegNet also uses perturbation in an iterative manner, in each training epoch and separately for each of the views, as detailed in Fig. 2. From the input data, 70% is selected, then 80% of this is processed by left-right and up-down flips, and 25% has 90°, 180°, 270° rotations applied. Random brightness and contrast changes are applied to 20% and 30% of the data respectively. Data perturbation is used both on labeled data and on the pseudo labels predicted from each classifier.

**Low Confidence Label Detection.** A badly performing view may negatively influence the whole framework in predicting the pseudo label process for the other two views. Uncertainty estimation can be tackled with agreement-based entropy-maps, or using pseudo labels for training, or computing uncertainty-weighted with a Bayesian deep network by adding dropout layers [23, 8]. Instead, we mark any unreliable pseudo labels as noisy [11, 22], thus temporarily removing low confidence labels from one view with the help of other two views. Retaining only high confidence pseudo labels boosts the overall performance. A simple adaptive denoising learning strategy is to calculate the overlap-based difference between each prediction in different views and record it for each pseudo label in each training epoch. The higher the label difference, the higher probability of it not being accepted. During each training epoch with prediction label  $(x, y_{pred})$ , our strategy detects and removes a specific number of high difference labels  $(x, y_{pseudo})$  raised by the other two views. More low confidence labels get detected at the beginning of the training iteration, and fewer towards the end., because the training process evolves from underfitting to overfitting. The number  $N(t)$  of removed labels is

$$N(t) = \begin{cases} 0.05(1 - \zeta)y, & 0 < t < 0.01(1 - \zeta)x \\ \frac{-y}{x}t + 0.06(1 - \zeta)y, & 0.01(1 - \zeta)x \leq t \leq 0.05(1 - \zeta)x \\ 0.01(1 - \zeta)y, & 0.05(1 - \zeta)x < t \leq x \end{cases} \quad (1)$$

where  $t$  is the current training iteration,  $\zeta$  is the difference/disagreement level,  $x$  is the total number of training iterations, and  $y$  is the total number of  $y_{pred}$ .

**Confidence Label Voting.** The uncertainty of the pseudo labels can sabotage expanding the training data, potentially influencing the whole framework performance. Luo [16] studied pyramid prediction network and uncertainty rectified pyramid consistency. Yu [27] proposed a Monte Carlo Dropout to generate an uncertainty map in student-teacher networks. Xia [23] proposed uncertainty

weighted label fusion. Unlike the student-teacher network or single encoder-decoder networks, or directly calculating the average maximum posterior probability of another two views, Tri-net [5] proposes a direct confidence estimate. In our algorithm, labeled training data and 20% of un-perturbed pseudo label data is considered as a validation set for each view to obtain confidence weights; thus two views jointly generate pseudo labels for the last view using confidence voting together after low confidence label detection for cross pseudo supervision [8].

### 2.3 Loss Function

The weight of pseudo-label-based semi-supervision loss compared with labeled-based supervised loss increases gradually during training, as the model becomes confident to generate and utilize pseudo labels [15, 19, 8]. The same low to high confidence has been explored in uncertainty-aware schemes [27]. TriSegNet enables coarse-to-fine model training using gradually more precise pseudo labels: in Stage 1 an overlap-based loss is used (Equation 2), whereas Stages 2 and 3 rely on a mixed overlap- and boundary-based loss (Equation 3). Overlap-based loss is more robust when the datasets are imbalanced (e.g. fewer True pixels). After the model the pseudo label from each view can be considered as a reliable annotation resource, a dual overlap-boundary-based loss is triggered, enabling the three views to segment more precisely [7].

$$Loss_{stage1} = \left( 1 - \frac{TP + 10^{-6}}{TP + \alpha FN + \beta FP + 10^{-6}} \right)^\gamma \quad (2)$$

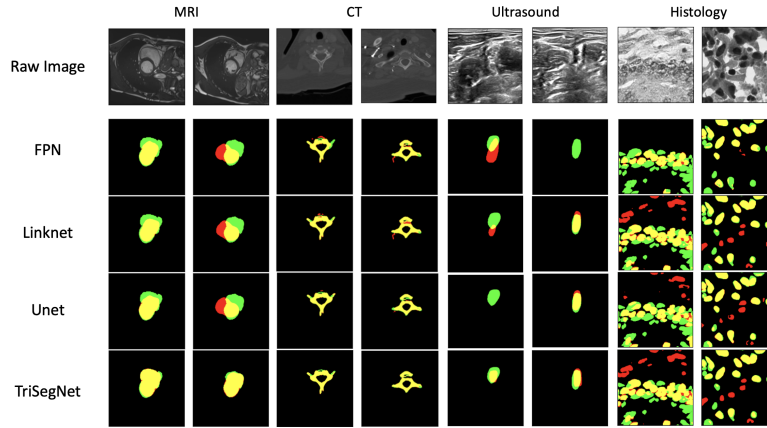
$$Loss_{stage2\&3} = \underbrace{\sum_{i,j=1}^{256} \sqrt{\|\nabla P_{x_{i,j}} + \nabla P_{y_{i,j}} + 10^{-6}\|}}_{Loss_{Boundary}} + \underbrace{\left\| \sum_{i,j=1}^{256} P_{i,j}(1 - G_{i,j})^2 \right\| + \left\| \sum_{i,j=1}^{256} (1 - P_{i,j})G_{i,j}^2 \right\|}_{Loss_{Overlap}} \quad (3)$$

The ground truth, and pseudo label are denoted as  $G, P \in [0, 1]^{256 \times 256}$ , where 1 and 0 denote the region of interest or the background on a  $256 \times 256$  image.

## 3 Experiments and Results

### 3.1 Datasets and Experimental Setup

Our experiments have been carried out on four public benchmark datasets: Ultrasound Nerve [12], CT Spine [24], MRI Cardiac [2], and Histology Nuclei [10]. Ultrasound, MRI and Histology images have been resized to  $256 \times 256$ , and CTs to  $512 \times 512$ . All four are pre-processed with data perturbation. The data has been split into labeled training data, test data and unlabeled raw training data.



**Fig. 3.** Sample qualitative results; Two images are selected from each test set where the first row illustrates the raw images. The rest of them illustrate the MS against GT where yellow, green, red, black represent TP, FN, FP and TN at pixel level.

The test data is always 10% of the set, selected randomly for each run. The training data is the remainder of the dataset; for separate experiments, 2%, 5%, 10% or 20% of this is available labeled, and the rest is available raw. The validation data is selected as a separate random 20% sample of the training set (both labeled not), and is never seen by the training process.

1. The Ultrasound Nerve Segmentation is a Kaggle Challenge [12]. Identifying nerve structures in ultrasound images is critical to inserting a patient’s pain management catheter. We use 5,635 images of size  $580 \times 420$ .
2. The MRI Cardiac Segmentation is from the Automated Cardiac Diagnosis MICCAI Challenge 2017 [2]. We use 1,203 images of size  $232 \times 256$ , from 150 patients.
3. The Histology Nuclei Segmentation is a Pan-Cancer histology dataset from the University of Warwick [10]. We use 7,901 images of size  $256 \times 256$ .
4. The CT Spine set includes 10 CTs from a CSI 2014 Segmentation Challenge [24], covering the entire thoracic and lumbar spine. We use 5,602 images of size  $512 \times 512$ .

Our Python 3.8.8 uses Tensorflow 2.6.0 [1] and CUDA 11.3, on four Nvidia GeForce RTX3090 GPUs. The runtimes averaged 2–3.5 hours on the MRI dataset, and 6–8 hours on the Ultrasound, CT and Histology datasets, including the data transfer, three training stages, inference and evaluation. Semi Supervised Learning for Medical Image Segmentation,<sup>1</sup> an online collection of implementations has been used in the baseline testing models and additional Dropout.

### 3.2 Evaluation and Results

Performance has been assessed using a wide range of evaluation metrics. These include the frequently reported similarity measures: Dice, IOU, accuracy, preci-

<sup>1</sup> <https://github.com/HiLab-git/SSL4MIS>

Experiments Under the Assumption of 2% Data as Labeled Data								
	Ultrasound Nerve		CT Spine		MRI Cardiac		Histology Nuclei	
	IOU	Sen	IOU	Sen	IOU	Sen	IOU	Sen
UNet	0.1628	0.2020	0.8657	0.9210	0.3888	0.8351	0.6574	0.7814
Linknet	0.0919	0.1280	0.8438	0.8962	0.1498	<b>0.9329</b>	0.6905	0.8219
FPN	0.1227	0.1320	0.8653	0.8990	0.4802	0.5143	0.6942	0.8284
<b>TriSegNet</b>	<b>0.2800</b>	<b>0.3678</b>	<b>0.9526</b>	<b>0.7024</b>	<b>0.9923</b>	0.8411	<b>0.6946</b>	<b>0.8293</b>
Experiments Under the Assumption of 5% Data as Labeled Data								
UNet	0.2762	0.3234	0.9207	0.9576	0.6805	0.8053	0.7075	0.5953
Linknet	0.2505	0.2885	0.9020	0.9519	0.6762	0.7988	0.7552	0.8645
FPN	0.2703	0.3093	0.9221	0.9521	0.7721	0.8351	0.7031	0.8333
<b>TriSegNet</b>	<b>0.3765</b>	<b>0.5090</b>	<b>0.9385</b>	<b>0.9660</b>	<b>0.8094</b>	<b>0.8880</b>	<b>0.7691</b>	<b>0.8757</b>
Experiments Under the Assumption of 10% Data as Labeled Data								
UNet	0.3554	0.4090	0.9320	0.9660	0.8492	<b>0.9253</b>	0.8012	0.8880
Linknet	0.3464	0.3991	0.9247	0.9527	0.7832	0.8641	0.7957	0.8862
FPN	0.2416	0.4866	0.8316	0.9721	0.8078	0.8591	0.8034	0.8973
<b>TriSegNet</b>	<b>0.4260</b>	<b>0.5789</b>	<b>0.9463</b>	<b>0.9714</b>	<b>0.8545</b>	0.9159	<b>0.8114</b>	<b>0.8981</b>
Experiments Under the Assumption of 20% Data as Labeled Data								
UNet	0.4352	0.5254	<b>0.9447</b>	<b>0.9699</b>	0.8984	0.9448	0.8104	0.9019
Linknet	0.4333	0.5237	0.9295	0.9678	0.8712	0.9258	0.8027	0.8971
FPN	0.3956	0.5953	0.7605	0.8920	0.8857	0.9307	0.8176	0.9094
<b>TriSegNet</b>	<b>0.4981</b>	<b>0.6528</b>	0.9337	0.9494	<b>0.9020</b>	<b>0.9459</b>	<b>0.8530</b>	<b>0.9244</b>

**Table 1.** Evaluation Results on Direct Comparison between TriSegNet and Existing Fully-Supervised Algorithms

sion, sensitivity, specificity, as well as difference measures: relative volume difference (RVD), Hausdorff distance (HD), average symmetric surface distance (ASSD). To penalise mislabelled areas and avoid overly inflated segmentation scores, we also evaluate the boundary match between the machine segmentation (MS) and the Ground Truth (GT), using the Directed Boundary Dice relative to GT ( $DBD_G$ ) and to MS ( $DBD_M$ ) and Symmetric Boundary Dice (SBD) [25]. Experiments first compare TriSegNet with fully-supervised baseline methods like UNet[18], LinkNet[4] and FPN[14]. Table 1 reports the quantitative results for each dataset, using 2%–20% of labeled data. The examples in Figure 3 illustrate these comparisons qualitatively.

Model	Dice	Acc	Pre	Rec/Sen	Spe	RVD	HD	ASSD	$DBD_G$	$DBD_M$	SBD
[19]	0.895	0.992	0.909	0.881	0.996	0.226	21.876	4.077	0.561	0.649	0.599
[21]	0.881	0.991	0.918	0.847	0.997	0.273	18.335	3.954	0.543	<b>0.651</b>	0.589
[28]	0.878	0.990	0.858	0.899	0.994	0.332	24.235	5.046	0.577	0.564	0.559
[27]	0.890	0.991	0.903	0.878	0.996	0.244	22.851	4.669	0.550	0.636	0.585
[20]	0.899	0.992	0.909	0.889	0.996	<b>0.205</b>	28.388	4.773	0.582	0.639	0.605
<b>TriSegNet</b>	<b>0.932</b>	<b>0.995</b>	<b>0.934</b>	<b>0.930</b>	<b>0.997</b>	0.208	<b>7.831</b>	<b>2.075</b>	<b>0.712</b>	0.611	<b>0.657</b>

**Table 2.** Direct comparison of TriSegNet with other algorithms, on the MRI Test Set

**Table 3.** Ablation Studies on Contributions of Architecture and Modules

Label Process	Dual Loss Design	Classifier A	Classifier B	Classifier C	IOU
		✓ × 2	✓		0.8724
			✓ × 2	✓	0.8739
		✓		✓ × 2	0.8641
✓		✓ × 3			0.8666
	✓	✓ × 3			0.8579
			✓ × 3		0.8598
✓			✓ × 3		0.8605
				✓ × 3	0.8619
✓				✓ × 3	0.8739
✓		✓	✓	✓	0.8787
	✓	✓	✓	✓	0.8841
✓	✓	✓	✓	✓	0.9020

To illustrate a wider range of evaluation measures with semi-supervised algorithms, there is only room to report on one dataset. The MRI Cardiac data has been chosen for this purpose. Table 2 documents the performance of TriSegNet against Tarvainen [19], Vu [21], Zhang [28], Yu [27], Verma [20] with UNet as backbone, showing that it outperforms previous methods under most of the considered metrics.

In order to assess the contribution of each of the components, specifically focused ablation experiments have been designed. They illustrate the essential role of models A, B and C being different by considering, instead of the three, two copies of one of them ( $A \times 2$ ,  $A \times 3$ , etc.). Table 3 illustrates that the IOU metric is negatively influenced by such choices, performing best when each of A, B and C is present in its own right.

## 4 Conclusion

Four medical datasets with different labeled/unlabeled assumptions have been used in the experiments. A series of experiments are designed including the comparison between TriSegNet and fully-supervised learning algorithms, as well as its comparison with other semi-supervised learning algorithms. Ablation studies justify the design decisions. Although a common Dice-based loss function is used for the initial stages of the training process, a bespoke boundary-overlap-based loss is used in the more advanced stages. This increases the confidence of the model in its predictions and hence the reliability of the pseudo labels it generates. Overall, TriSegNet demonstrates promising performance most evaluation metrics, showing great potential in semi-supervised learning for general medical image segmentation.

## References

1. Martín Abadi and etc. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
2. Olivier Bernard et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37(11):2514–2525, 2018.
3. Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.
4. Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing*, pages 1–4. IEEE, 2017.
5. Dong-Dong Chen et al. Tri-net for semi-supervised deep learning. In *International Joint Conferences on Artificial Intelligence*, 2018.
6. Liang-Chieh Chen et al. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *European Conference on Computer Vision*, pages 695–714. Springer, 2020.
7. Xu Chen et al. Learning active contour models for medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11640, 2019.
8. Xiaokang Chen et al. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.
9. Jeff Donahue et al. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.
10. Jevgenij Gamper et al. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In *European Congress on Digital Pathology*, pages 11–19. Springer, 2019.
11. Jinchu Huang et al. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3326–3334, 2019.
12. Kaggle. Ultrasound nerve segmentation. <https://www.kaggle.com/c/ultrasound-nerve-segmentation>.
13. Zhanghan Ke et al. Guided collaborative training for pixel-wise semi-supervised learning. In *European conference on computer vision*, pages 429–445. Springer, 2020.
14. Seung-Wook Kim et al. Parallel feature pyramid network for object detection. In *Proceedings of the European Conference on Computer Vision*, pages 234–250, 2018.
15. Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.
16. Xiangde Luo et al. Efficient semi-supervised gross target volume of nasopharyngeal carcinoma segmentation via uncertainty rectified pyramid consistency. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 318–329. Springer, 2021.
17. Siyuan Qiao et al. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision*, pages 135–152, 2018.
18. 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.

19. Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1195–1204, 2017.
20. Vikas Verma et al. Interpolation consistency training for semi-supervised learning. In *International Joint Conference on Artificial Intelligence*, pages 3635–3641, 2019.
21. Tuan-Hung Vu et al. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.
22. Z Wang, Z Zhang, and I Voiculescu. RAR-U-Net: a residual encoder to attention decoder by residual connections framework for spine segmentation under noisy labels. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2021.
23. Yingda Xia et al. 3d semi-supervised learning with uncertainty-aware multi-view co-training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3646–3655, 2020.
24. Jianhua Yao, Joseph E Burns, and etc. Detection of vertebral body fractures based on cortical shell unwrapping. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 509–516. Springer, 2012.
25. Varduhi Yeghiazaryan and Irina D Voiculescu. Family of boundary overlap metrics for the evaluation of medical image segmentation. *Journal of Medical Imaging*, 5(1):015006, 2018.
26. Xinge You et al. Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. *Pattern recognition*, 44(10-11):2314–2324, 2011.
27. Lequan Yu, Shujun Wang, Xiaomeng Li, Chi-Wing Fu, and Pheng-Ann Heng. Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 605–613. Springer, 2019.
28. Yizhe Zhang et al. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *International conference on medical image computing and computer-assisted intervention*, pages 408–416. Springer, 2017.

## A Algorithm of TriSegNet

The training of TriSegNet consists of four stages which is briefly illustrated in Algorithm 1.

## B The CNN Architecture of Multi-View Learning

To properly encourage the differences of three views of feature learning on dense prediction, not only the data feed and initialization of parameters, but three different advanced CNN are proposed in TriSegNet. We utilize three different techniques for CNN i.e. skip connection, efficiently pass feature information through residual learning, and multi-scale feature learning. The parameters of three classifier is briefly illustrated in Table 4 and the source code is in <sup>2</sup>.

## C Evaluation Methods, Qualitative, and Quantitative Results

Table 2 reports the TriSegNet performance direct comparison with other algorithms with several strict and novel quantitative evaluation metrics to which the boundaries of the machine segmentation(MS) match those of the ground truth(GT), using the Directed Boundary Dice relative to GT ( $DBD_G$ ), Directed Boundary Dice relative to MS ( $DBD_M$ ) and Symmetric Boundary Dice (SBD).

In a von Neumann neighbourhood  $N_x$  of each pixel  $x$  on the boundary  $\partial G$  of the ground truth,

$$DBD_G = DBD(G, M) = \frac{\sum_{x \in \partial G} \text{Dice}(N_x)}{|\partial G|} \quad (4)$$

$$DBD_M = DBD(M, G) = \frac{\sum_{x \in \partial M} \text{Dice}(N_x)}{|\partial M|} \quad (5)$$

$$SBD = \frac{\sum_{x \in \partial G} \text{DSC}(N_x) + \sum_{y \in \partial M} \text{DSC}(N_y)}{|\partial G| + |\partial M|} \quad (6)$$

where Dice is  $\text{Dice}(N_x) = \frac{2|G(N_x) \cap M(N_x)|}{|G(N_x)| + |M(N_x)|}$ . The symmetric average is being brought down by  $DBD_G$  when the latter features isolated areas of false negative labels. These measures penalise mislabelled areas in the machine segmentation.

Some of example qualitative results on MRI Cardiac test set are briefly sketched in Figure 4. Eight images are selected from MRI test set where the first row illustrate raw images. The rest of them illustrate the MS by each semi-supervised algorithm against GT where yellow, green, red, black represent true

<sup>2</sup> [https://github.com/qubvel/segmentation\\_models/tree/master/segmentation\\_models](https://github.com/qubvel/segmentation_models/tree/master/segmentation_models)

---

**Algorithm 1:** Training Stage of TriSegNet for Medical Image Segmentation
 

---

**Input:** A batch of  $(x, y)$  from labeled dataset  $\mathcal{L}$ , unlabeled dataset  $\mathcal{U}$ , or test dataset  $\mathcal{T}$ .  $DA$ ,  $LACLE$ , and  $CLV$  are label processing approaches Data Augmentation, Low Confidence Label Editing, and Confident Label Voting, respectively.

**Output:** Three trained high-level feature learning classifiers.  $f_n, n = 1, 2, 3$

**Stage one:** Initialization

$\theta_1, \theta_2, \theta_3 \leftarrow$  initial parameters of classifiers  $f_1, f_2, f_3$ , and the loss function  $L_{Supervision}$ .

**while**  $n \leftarrow [1, 2, 3]$  (3 classifiers) **do**  
      $(x, y_{gt})$  sampled from  $DA(\mathcal{L})$  with augmentation;  
     Generate prediction  $y_{pred} = f_n(x)$  ;  
     Calculate loss  $L_{Supervision}$  with  $y_{pred} = y_{gt}$  ;  
     Update  $\theta_n \leftarrow \theta_n - \Delta L$  ;

**end**

**Stage two:** Classifiers Training with Pseudo Label Processing

$\alpha_1, \alpha_2, \alpha_3 \leftarrow$  initial confidence weight of classifiers  $f_1, f_2, f_3$  ;

**while**  $n \leftarrow [1, 2, 3], i \leftarrow [1...5]$  (5 Iterations) **do**  
      $x$  sampled from  $DA(\mathcal{U})$  with augmentation ;  
     Generate pseudo label for Classifier  $f_n$   
      $y_{n\_pseudo} = CLV(LACLE(f_{n-1}(x), LACLE(f_{n+1}(x))))$  ;  
     Calculate loss  $L_{SemiSupervision}$  with  $y_{pred} = y_{n\_pseudo}$  ;  
     Update  $\theta_n \leftarrow \theta_n - \Delta L$  ;  
     Update  $\alpha_n \leftarrow$  with  $L$  by evaluation dataset ;

**end**

**Stage three:** One low confidence Classifier Training

**while**  $n \leftarrow [1, 2, 3], i \leftarrow [1...5]$  **do**  
     **If** Network  $f_n$  is with the lowest  $\alpha_n$  **Then**  
      $(x, y_{gt})$  sampled from  $DA(\mathcal{L})$  ;  
      $(x, y_{n\_pseudo})$  sampled from  $DA(\mathcal{U})$  ;  
     Generate pseudo label for Classifier  $f_n$   
      $y_{n\_pseudo} = CLV(LACLE(f_{n-1}(x), LACLE(f_{n+1}(x))))$  ;  
     Calculate loss  $L$  with  $y_{pred} = y_{n\_pseudo}$  ;  
     Update  $\theta_n \leftarrow \theta_n - \Delta L_{SemiSupervision}$  ;  
     Update  $\alpha_n \leftarrow$  with  $L$  by evaluation dataset ;

**end**

---

Model	Total Params	Trainable params	Non-trainable params
Classifier A	32,561,120	32,513,562	47,558
Classifier B	20,323,985	20,317,169	47,558
Classifier C	17,594,453	17,592,149	2,304

**Table 4.** The Computation Cost Information of Three Classifier

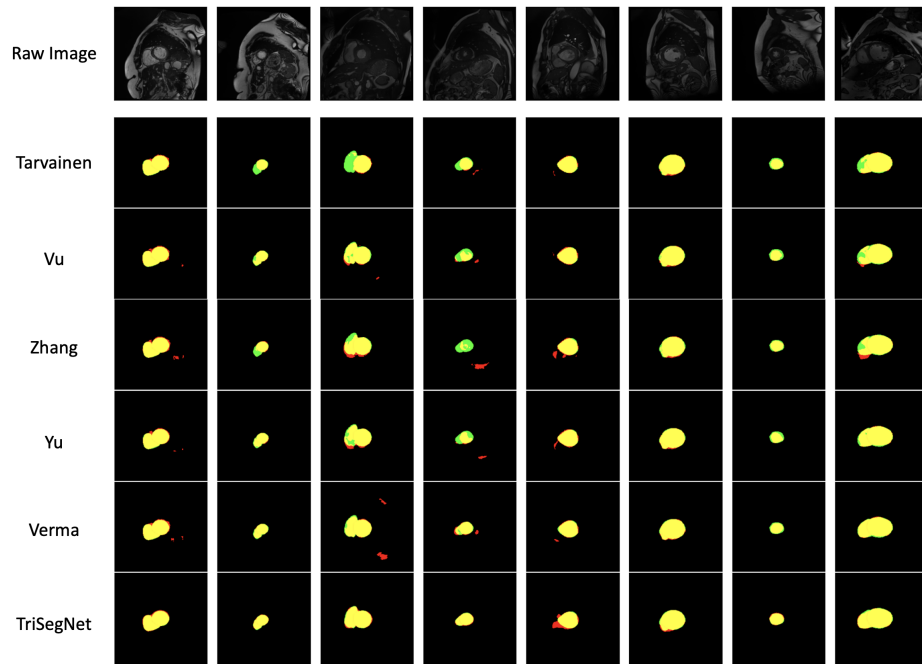


Fig. 4. Sample Qualitative Results on MRI Cardiac Test Set

positive, false negative, false positive and true negative at pixel level. The proposed method shows less false positive and false negative pixels, and significantly low HD as well, because the TriSegNet is beneficial with different views of high-level pixel-level classifier and proposed mixed boundary- and overlap-based loss function.