Medical Vision Seminar

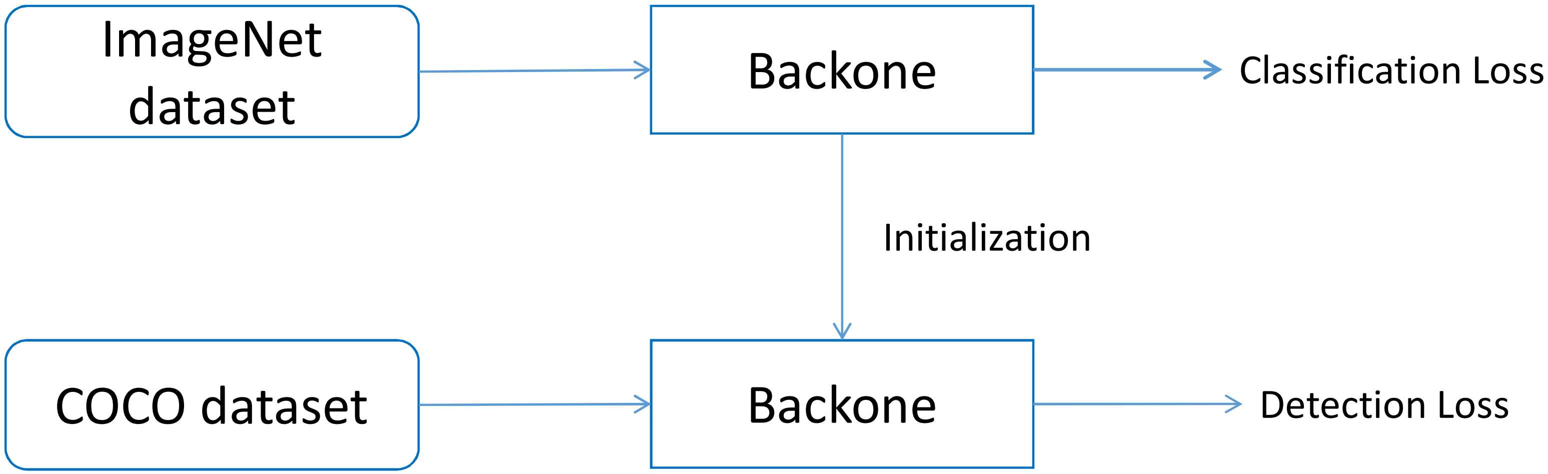
——Wei Lou

（NIPS2020） Rethinking Pre-training and Self-training

—— Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin,Yin Cui,
Hanxiao Liu, Ekin D. Cubuk, Quoc V. Le Johns
Google Research, Brain Team

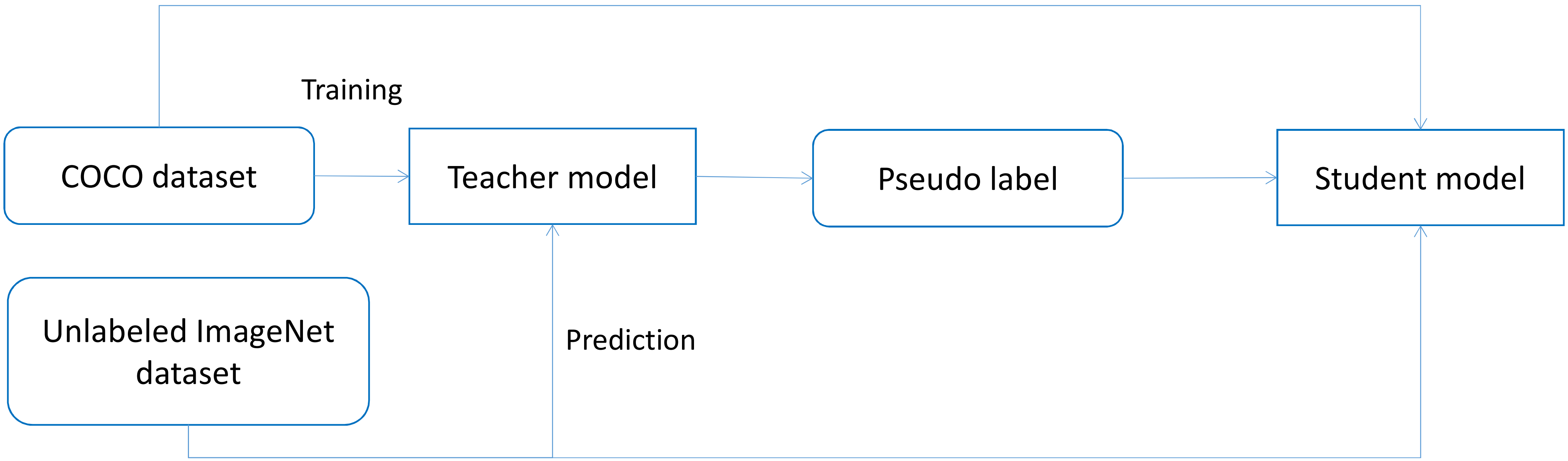
1. Introduction

Pre-training: A model , pre-trained on one dataset to help the training initialization on other datasets. For example, pre-train the backbones of object detection and segmentation models on ImageNet classification task or self-supervised task.



1. Introduction

Self-training: A simple semi-supervised method, a teacher model is trained on the labeled data (COCO dataset), then the teacher model generates pseudo labels on unlabeled data (ImageNet dataset). Finally, a student model is trained to optimize the loss on both labeled or pseudo data.



1.1 Why investigate them?

Pre-training fails in many cases:

1. ImageNet pre-training does not improve detection accuracy on COCO dataset, even hurts the performance if using full labeled data.
2. ImageNet pre-training is not necessary for semantic segmentation with CityScapes dataset if aggressive data augmentation is applied.
3. ImageNet pre-training does not improve medical image classification tasks.

Self-training shows good progress:

Recently, self-training has shown great success in classification / detection / segmentation tasks. However, they only study self-training in isolation without a comparison with ImageNet pre-training.

1.2 Goals:

- Study pre-training in detail (strong data augmentation, different pre-training methods (supervised/self-supervised), different pre-trained checkpoint qualities).
- Study the generality and scalability of self-training.
- Compare the performance of ImageNet pre-training and self-training.

3. Method

3.1 Data augmentation:

Augment-S1: Flips and Crops (Weakest)
Augment-S2: **AutoAugment**, Flips and Crops (Third strongest)
Augment-S3: Large Scale Jittering, AutoAugment, Flips and Crops (Second strongest)
Augment-S4: Large Scale Jittering, **RandAugment**, Flips and Crops (Strongest)

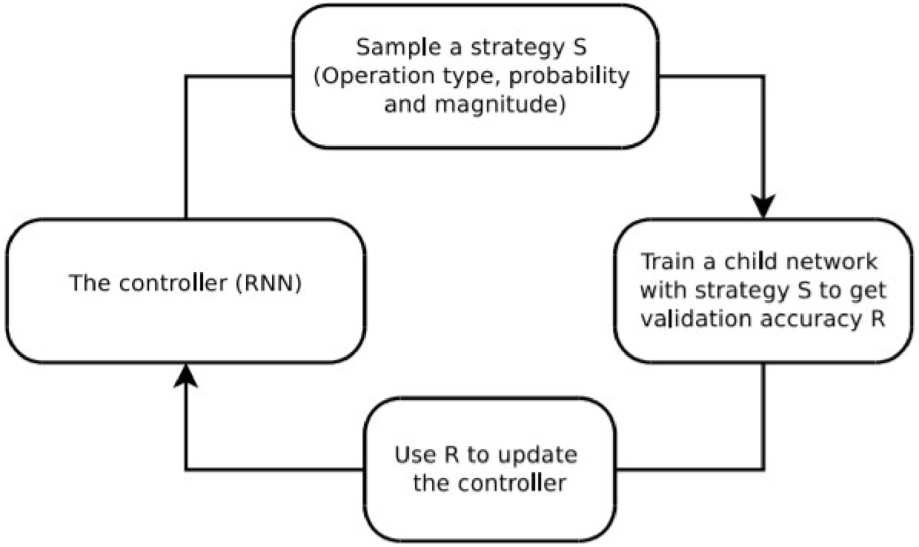


Fig.1 Auto Augment, reprint from [1]

[1] CVPR 2019 <AutoAugment:Learning Augmentation Strategies from Data>

3.2 Pre-training: different checkpoints

Rand Init: Model initialized with random weights (Weakest)
ImageNet Init: Model initialized with ImageNet pre-trained checkpoint (84.5% top-1) (Second strongest)
ImageNet++ Init: Model initialized with higher performing ImageNet pre-trained checkpoint (86.9% top-1) (Strongest)

3.3 Self-training:

A teacher model is trained on the labeled data (COCO dataset), then the teacher model generates pseudo labels on unlabeled data (ImageNet dataset). Finally, a student model is trained to optimize the loss on both labeled or pseudo data.

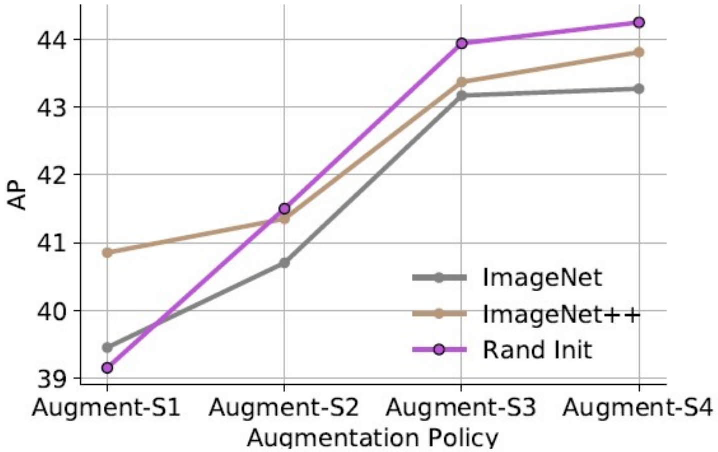
4. Experiments

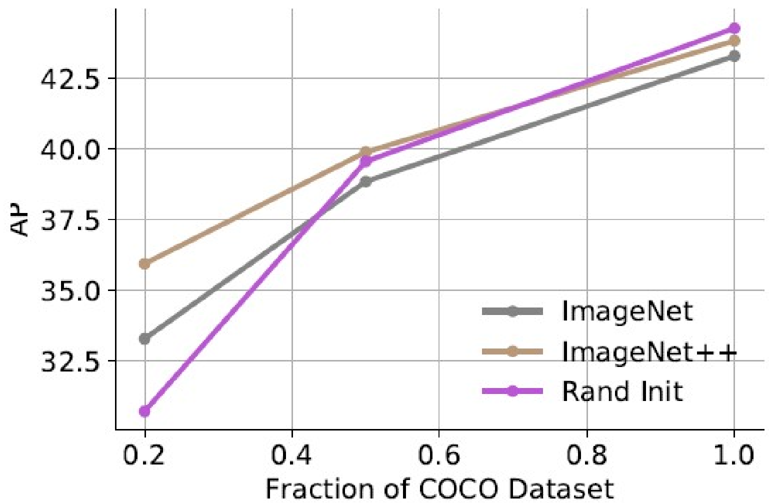
4.1 The effects of augmentation and labeled dataset size on pre-training

Setting: Use ImageNet pre-training and vary the COCO dataset size, data augmentation strengths, pre-trained model qualities.

Finding 1: Pre-training hurts performance when stronger data augmentation is used.

Finding 2: More labeled data diminishes the value of pre-training / Better checkpoint does correlate with the performance in low data regime.





4.2 The effects of augmentation and labeled dataset size on self-training

Setting: Use self-training and the same detection backbone with different COCO dataset size, data augmentation strengths. Use the same ImageNet dataset without labels.

Finding 3: Self-training helps in high data/strong augmentation regimes, even when pre-training hurts.
Finding 4: Self-training works across dataset sizes and is additive to pre-training.

Setup	Augment-S1	Augment-S2	Augment-S3	Augment-S4
Rand Init	39.2	41.5	43.9	44.3
ImageNet Init	(+0.3) 39.5	(-0.7) 40.7	(-0.8) 43.2	(-1.0) 43.3
Rand Init w/ ImageNet Self-training	(+1.7) 40.9	(+1.5) 43.0	(+1.5) 45.4	(+1.3) 45.6

Setup	20% Dataset	50% Dataset	100% Dataset
Rand Init	30.7	39.6	44.3
Rand Init w/ ImageNet Self-training	(+3.4) 34.1	(+1.8) 41.4	(+1.3) 45.6
ImageNet Init	33.3	38.8	43.3
ImageNet Init w/ ImageNet Self-training	(+2.7) 36.0	(+1.7) 40.5	(+1.3) 44.6
ImageNet++ Init	35.9	39.9	43.8
ImageNet++ Init w/ ImageNet Self-training	(+1.3) 37.2	(+1.6) 41.5	(+0.8) 44.6

4.3 Self-supervised pre-training also hurts when self-training helps in high data/strong augmentation regimes

Setting: Choose a SimCLR checkpoint trained on ImageNet dataset. Compare the detection performance in high data/strong augmentation regimes.

Setup	COCO AP
Rand Init	41.1
ImageNet Init (Supervised)	(-0.7) 40.4
ImageNet Init (SimCLR)	(-0.7) 40.4
Rand Init w/ Self-training	(+0.8) 41.9

Finding 5: Both pre-trained models hurts detection accuracy but self-training still improve the performance.

4.4 Exploring the limits of self-training and pre-training

Setting:

COCO detection: Choose SpineNet as backbones, OpenImage Dataset as unlabeled dataset and augment-S3.

PASCAL VOC Semantic Segmentation: Choose NAS-FPN architecture with EfficientNet-B7 and EfficientNet-L2 as the backbone architectures. Combine the ImageNet ++ pre-training, self-training and augment-S4. JFT datasets, COCO datasets.

Model	# FLOPs	# Params	AP (val)	AP (test-dev)
AmoebaNet+ NAS-FPN+AA (1536)	3045B	209M	50.7	—
EfficientDet-D7 (1536)	325B	52M	52.1	52.6
SpineNet-143† (1280)	524B	67M	50.9	51.0
SpineNet-143 (1280) w/ Self-training	524B	67M	(+1.5) **52.4**	(+1.6) **52.6**
SpineNet-190† (1280)	1885B	164M	52.6	52.8
SpineNet-190 (1280) w/ Self-training	1885B	164M	(+1.6) **54.2**	(+1.5) **54.3**

Model	Pre-trained	# FLOPs	# Params	mIOU (val)	mIOU (test)
ExFuse †	ImageNet, COCO			85.8	87.9 ‡
DeepLabv3+	ImageNet	177B		80.0	—
DeepLabv3+	ImageNet, JFT, COCO	177B		83.4	—
DeepLabv3+ †	ImageNet, JFT, COCO	3055B		84.6	89.0 ‡
Eff-B7	ImageNet++	60B	71M	85.2	—
Eff-B7 w/ Self-training	ImageNet++	60B	71M	(+1.5) **86.7**	—
Eff-L2	ImageNet++	229B	485M	88.7	—
Eff-L2 w/ Self-training	ImageNet++	229B	485M	(+1.3) **90.0**	**90.5**

Self-training without ImageNet ++: mIOU 41.5

5. Discussion

5.1 Rethinking pre-training

There is limitation of learning universal representations from both classification and self-supervised tasks. Pre-training is not aware of the task of interest and can fail to adapt. For example, good features for ImageNet may discard positional information which is needed for COCO.

5.2 Joint-learning may achieve better performance.

Joint-learning, ImageNet classification is trained jointly with COCO object detection using the same backbone.

Setup	Sup. Training	w/ Self-training	w/ Joint Training	w/ Self-training w/ Joint Training
Rand Init	30.7	(+3.4) 34.1	(+2.9) 33.6	(+4.4) 35.1
ImageNet Init	33.3	(+2.7) 36.0	(+0.7) 34.0	(+3.3) 36.6

5.3 Task-alignment
For PASCAL segmentation task, augmented data hurts the accuracy because they are noisy data. So noisy data (data augmentation) and un-targeted data (ImageNet) may worse than targeted pseudo labels.

Setup	train	train + aug	train + aug w/ Self-training
ImageNet Init w/ Augment-S1	83.9	(+0.8) 84.7	(+1.7) 85.6
ImageNet Init w/ Augment-S4	85.2	(-0.4) 84.8	(+1.5) 86.7

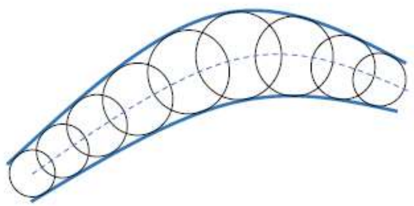
5.4 The scalability and flexibility of self-training
1. First, in terms of flexibility, self-training works well in every setup that we tried: low data regime, high data regime, weak data augmentation and strong data augmentation. Self-training also is effective with different architectures (ResNet, EfficientNet,SpineNet, FPN, NAS-FPN), data sources (ImageNet, OID, PASCAL, COCO) and tasks (ObjectDetection, Segmentation).
2. Self-training works well even when pre-training fails but also when pre-training succeeds.
3. In terms of scalability, self-training proves to perform well as we have more labeled data and better models.

（CVPR2020） Deep Distance Transform for Tubular Structure Segmentation in CT Scans

—— Yan Wang, Xu Wei, Fengze Liu, Jieneng Chen,
Johns Hopkins University, University of California San Diego, Tongji University

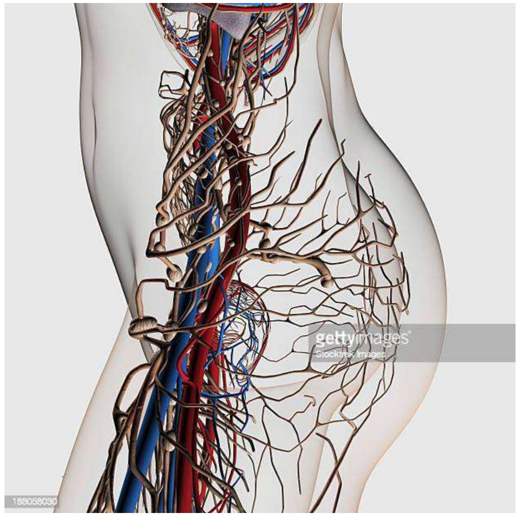
1. Introduction

1.1 Tubular structure segmentation



Tubular shape

Pancreatic duct (胰腺管)



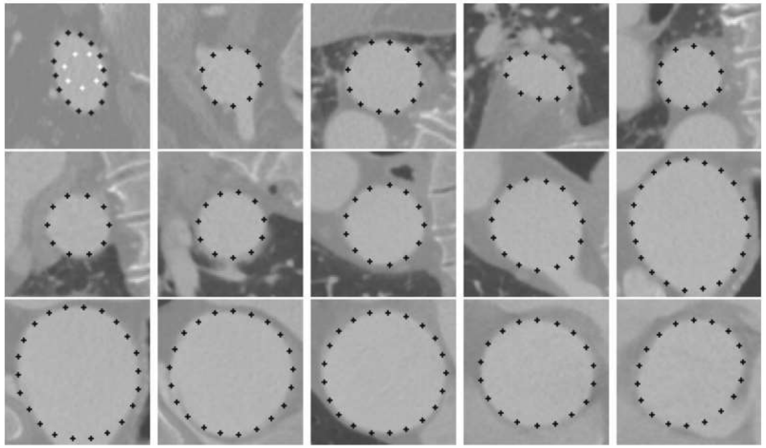
Vessels

Geometry feature: A tubular structure usually has a cylinder-like (柱状) shape which can be well represented by its skeleton and cross-sectional radius.

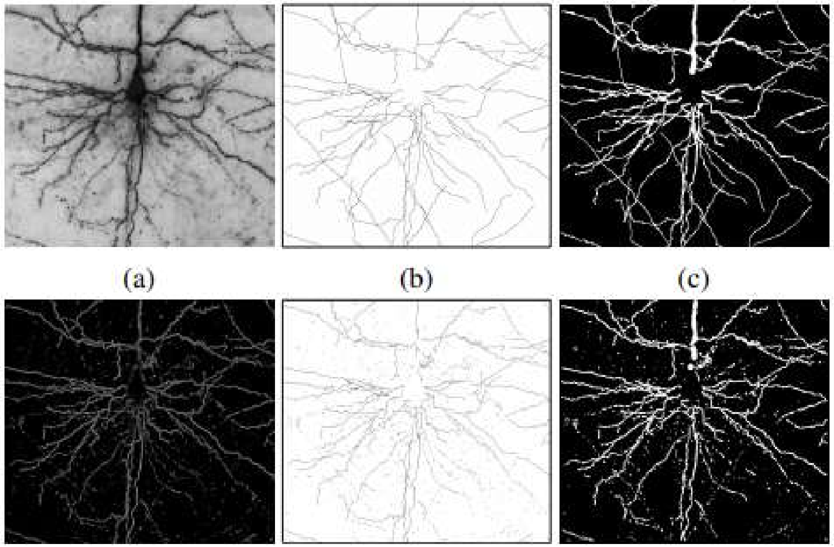
2. Related Work

2.1 Geometry-based Method

1. Contour-based methods:



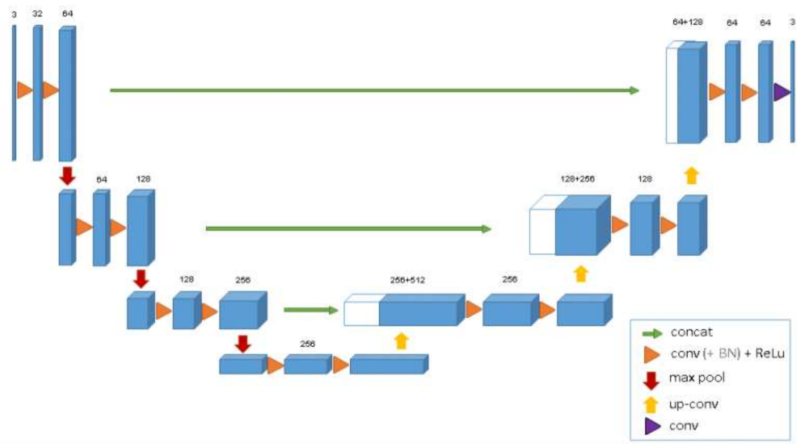
2. Centerline based methods



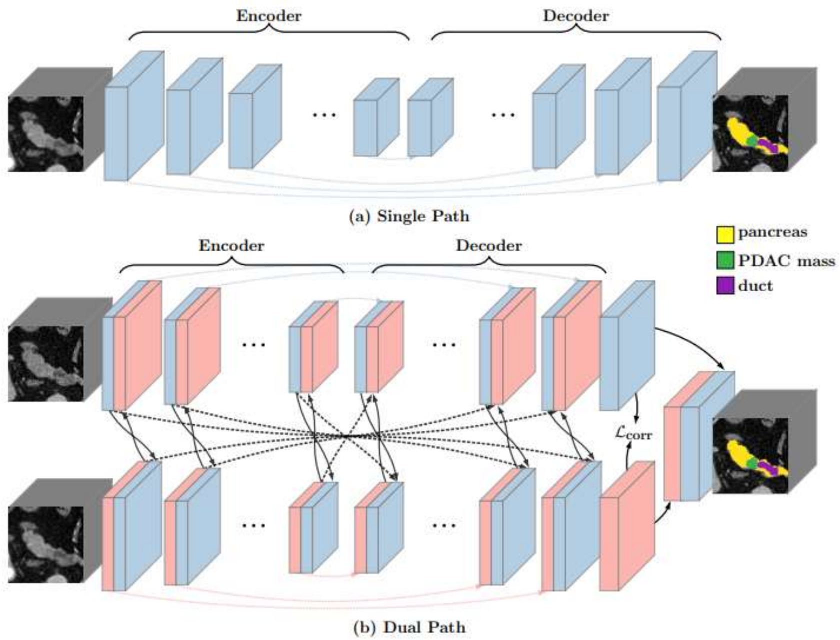
(a) (b) (c)

(d) (e) (f)

2.2 Learning-based Method

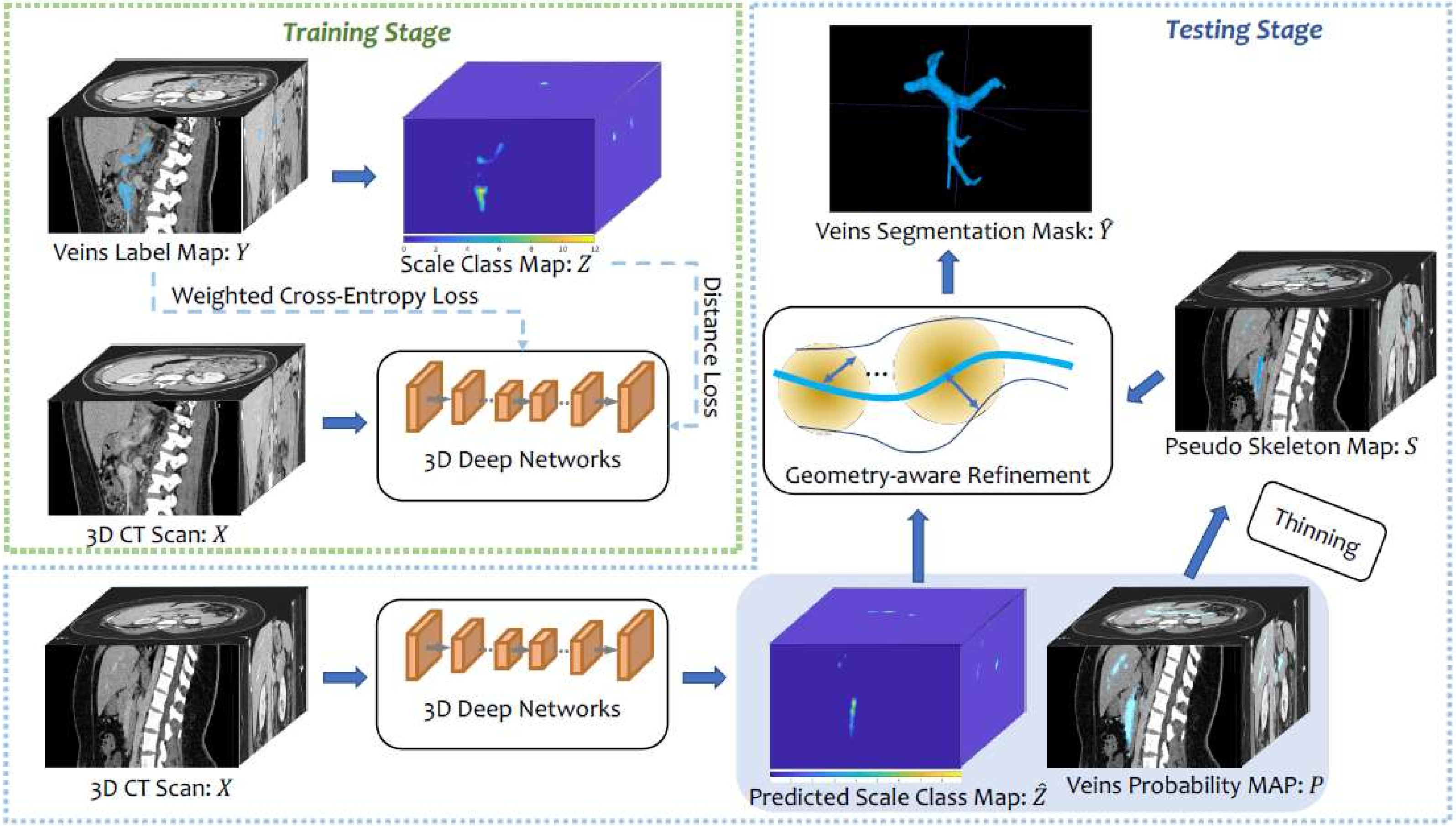


3D-Unet



HPN

3. Method



Training Stage: Voxel classification / Distance map
Testing Stage: Refinement with distance map

3.1 Distance map (Distance transform for tubular structure)

Binary segmentation: Tubular voxel $y_{\mathbf{v}} = 1$

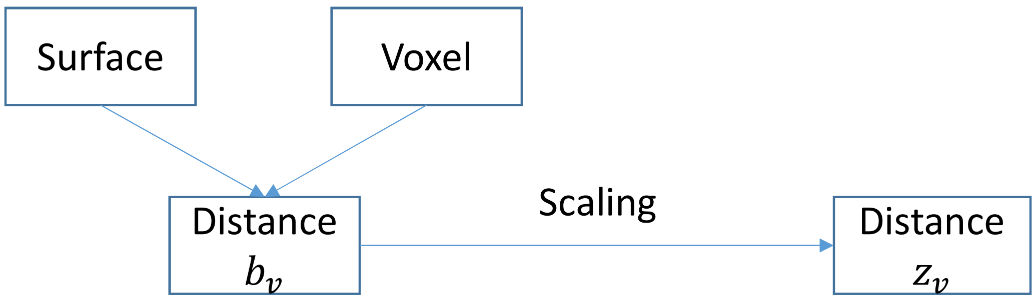
Tubular surface: $C_V = \{\mathbf{v} |\, y_{\mathbf{v}} = 1, \exists\, \mathbf{u} \in \mathcal{N}(\mathbf{v}), y_{\mathbf{u}} = 0\},$ (1)

Distance map D:
$$d_{\mathbf{v}} = \begin{cases} \min_{\mathbf{u} \in C_V} \|\mathbf{v} - \mathbf{u}\|_2, & \text{if } y_{\mathbf{v}} = 1 \\ 0, & \text{if } y_{\mathbf{v}} = 0 \end{cases}. \quad (2)$$

Each d is the distance between a voxel to its closest surface voxel. Because training a deep network directly for regression is relatively unstable, since outliers, i.e., the commonly existed annotation errors for medical images. So it's more reliable to form a classification task.

b_v is quantized into K bins by round to the nearest integer.

$$z_{\mathbf{v}} \in \{0, \ldots, K\}$$



3.2 Network Training

Data: 3D CT scan X / Ground truth label map Y / Scaled distance map Z

3.2.1 Weighted cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\sum_{\mathbf{v}\in V}\Big(\beta_p y_{\mathbf{v}} \log p_{\mathbf{v}}(\mathbf{W}, \mathbf{w}_{\text{cls}}) + \beta_n (1-y_{\mathbf{v}}) \log\big(1 - p_{\mathbf{v}}(\mathbf{W}, \mathbf{w}_{\text{cls}})\big)\Big), \qquad (3)$$

p_v: class label probability

$\beta_p = \frac{0.5}{\sum_{\mathbf{v}} y_{\mathbf{v}}}$ The number of positive / negative voxels

$\beta_n = \frac{0.5}{\sum_{\mathbf{v}} (1-y_{\mathbf{v}})}$

3.2.2 Weighted cross-entropy loss:

$$\mathcal{L}_{\text{dis}} = -\beta_p \sum_{\mathbf{v}\in V}\sum_{k=1}^{K}\Big(\mathbf{1}(z_{\mathbf{v}} = k)\Big(\log g_{\mathbf{v}}^{k}(\mathbf{W}, \mathbf{w}_{\text{dis}}) + \lambda\omega_{\mathbf{v}} \log\big(1 - \max_{l} g_{\mathbf{v}}^{l}(\mathbf{W}, \mathbf{w}_{\text{dis}})\big)\Big)\Big), \qquad (4)$$

First term: Softmax loss, g_v is the propability

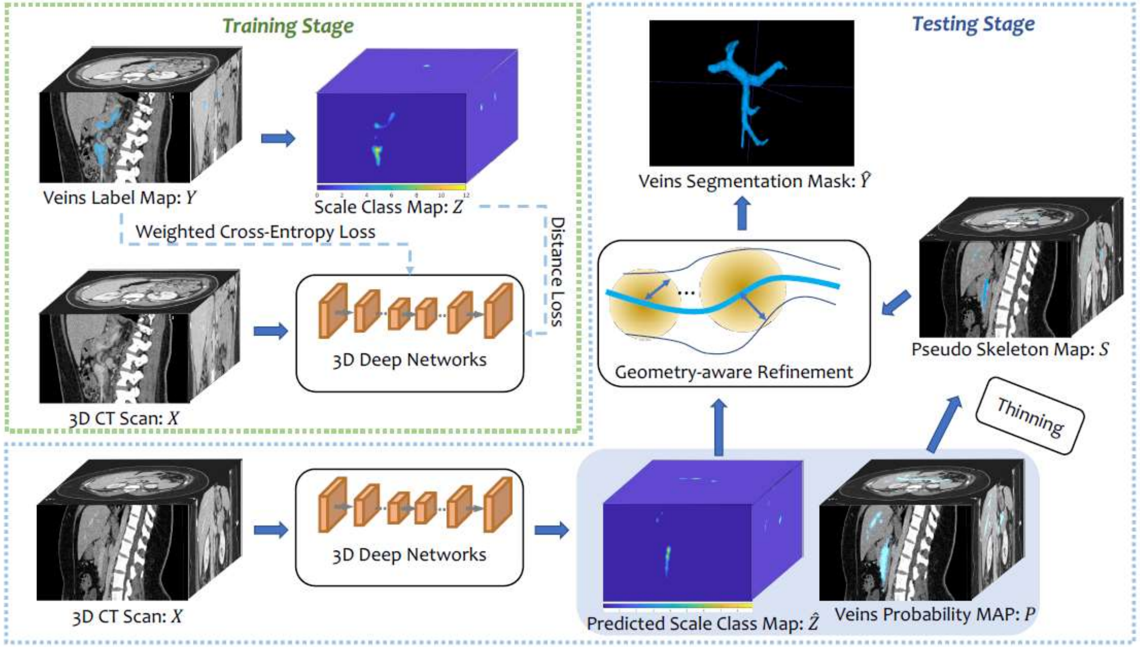
Second term:

$$\omega_{\mathbf{v}} = \frac{|\arg\max_l g_{\mathbf{v}}^{l}(\mathbf{W}, \mathbf{w}_{\text{dis}}) - z_{\mathbf{v}}|}{K}.$$

w_v: Distance of the predicted scale to the ground truth scale

3.3 Geometry-aware Refinement

Data: p_v : class predicted map, g_v: scaled distance map

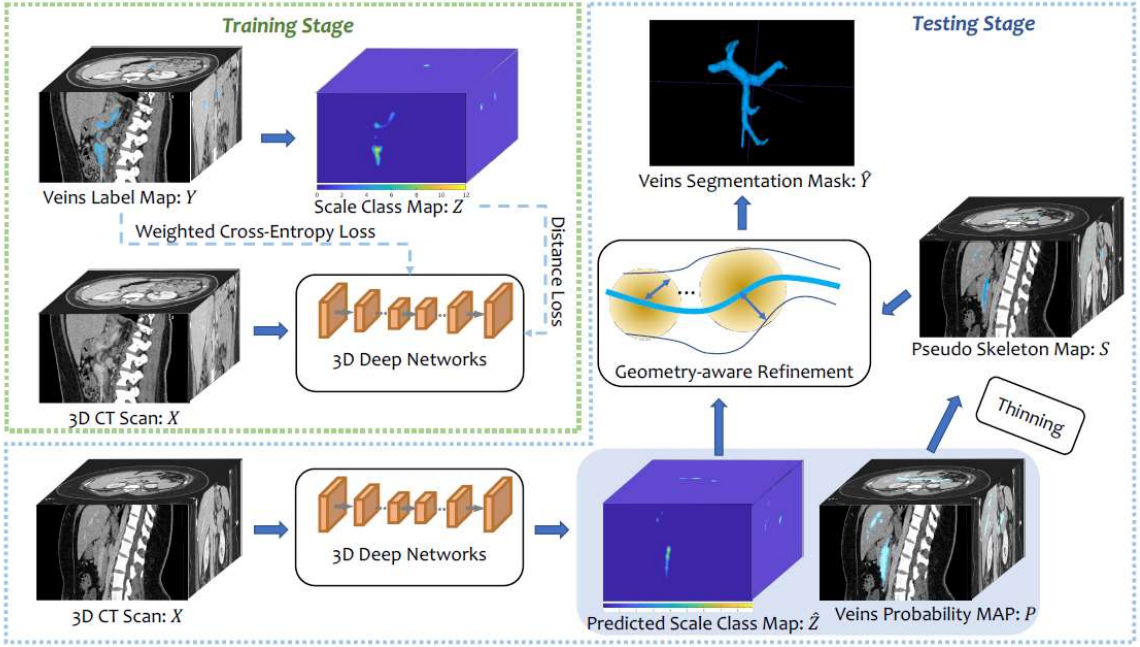


3.3.1 Pseudo skeleton generation

The probability map is thinned by thresholding it to generate a binary pseudo skeleton map S.

3.3 Geometry-aware Refinement

Data: p_v : class predicted map, g_v: scaled distance map



3.3.2 Shape reconstruction

1. For each voxel, get its predicted scale (radius).
2. Reconstruct the shape from:

$$\mathbf{v} \in \bigcup_{\mathbf{u}\in\{\mathbf{u}'|s_{\mathbf{u}'}>0\}} B(\mathbf{u}, \hat{z}_{\mathbf{u}})$$

B: ball centered at u with radius $\hat{z}_{\mathbf{u}}$

3. The quantized scale leads to an non-smooth surface: Fit a Guassian kernel to soften each ball and obtain a soft Reconstructed shape

$$\tilde{y}^s_{\mathbf{v}} = \sum_{\mathbf{u}\in\{\mathbf{u}'|s_{\mathbf{u}'}>0\}} c_{\mathbf{u}}\Phi(\mathbf{v};\mathbf{u},\mathbf{\Sigma}_{\mathbf{u}}),$$

Φ(.) multivariate normal distribution, u is the mean and $\sum$u is the co-variance matrix: according to 3-sigma rule

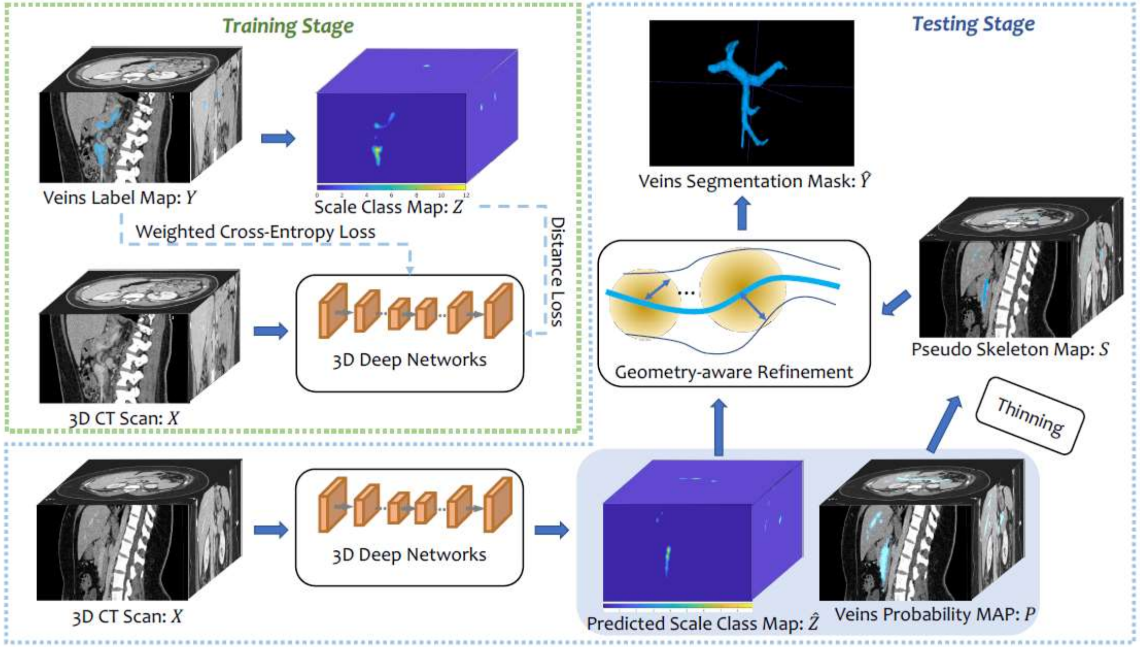
$$\mathbf{\Sigma}_{\mathbf{u}} = (\tfrac{\hat{z}_{\mathbf{u}}}{3})^2 \boldsymbol{I}$$

c_u is a normalization factor:

$$c_{\mathbf{u}} = \sqrt{(2\pi)^3 \det(\Sigma_{\mathbf{u}})}.$$

3.3 Geometry-aware Refinement

Data: p_v : class predicted map, g_v: scaled distance map



3.3.3 Segmentation refinement

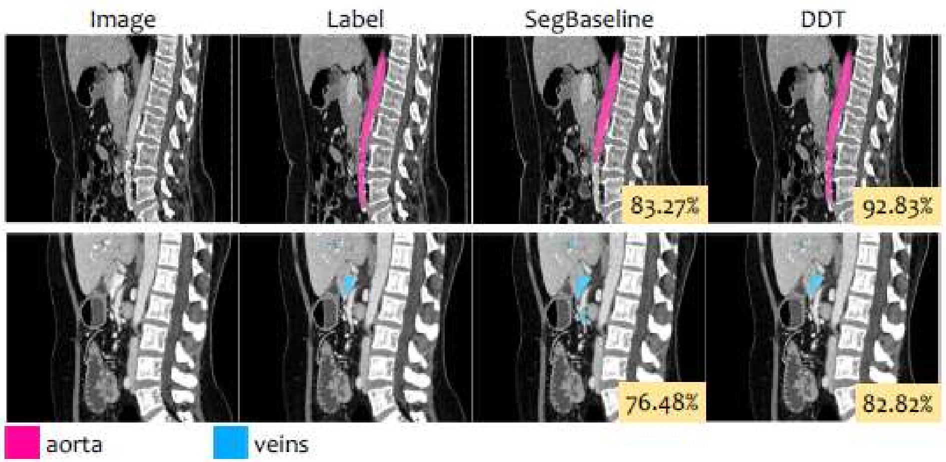
Apply the soft reconstructed shape to refine the segmentation map.

$$\tilde{y}_{\mathbf{v}}^{r} = \sum_{\mathbf{u} \in \{\mathbf{u}' | s_{\mathbf{u}'} > 0\}} p_{\mathbf{u}} c_{\mathbf{u}} \Phi(\mathbf{v}; \mathbf{u}, \Sigma_{\mathbf{u}}).$$

4. Experiments

4.1 Datasets: 3D CT scan

5 segmentation datasets: An dataset used in PDAC [1]; Three tubular structure datasets created themselves;
Hepatic vessels (肝脏血管) dataset in Medical Segmentation Decathlon (MSD) challenge.



[1] Hyper-pairing network for multi-phase pancreatic ductal adenocarcinoma segmentation. In Proc.MICCAI, 2019.

4.2 Results on PDAC segmentation datasets (Dice)

Compare with the SOTA:

Methods	Phase	Backbone Networks	
		3D-UNet	ResDSN
SegBaseline [46]	V	40.25 ± 27.89	49.81 ± 26.23
Multi-phase HPN [46]	A+V	44.93 ± 24.88	$\mathbf{56.77 \pm 23.33}$
DDT (Ours)	V	$\mathbf{58.20 \pm 23.39}$	55.97 ± 24.76

Ablation study, DDT: Deep distance transform; GAR: geometry-aware refinement

Method	Average DSC (%)
SegBaseline [46]	49.81
SegfromSkel	51.88
DDT $\lambda = 0$, w/o GAR	52.73
DDT $\lambda = 0$, w/ GAR	54.70
DDT $\lambda = 1$, w/o GAR	53.69
DDT $\lambda = 1$, w/ GAR	**55.97**

4.3 Results on three tubular segmentation datasets (Dice)

Backbone	Methods	Aorta		Veins		Pancreatic duct	
		Average DSC ↑	Mean Surface Distance ↓	Average DSC ↑	Mean Surface Distance ↓	Average DSC ↑	Mean surface Distance ↓
3D-HED [24]	SegBaseline	90.85	1.15	73.57	5.13	46.43	7.06
	DDT	92.94	0.82	76.20	**3.78**	54.43	4.91
3D-UNet [12]	SegBaseline	92.01	0.94	71.57	4.46	56.63	3.64
	DDT	**93.30**	**0.61**	75.59	4.07	**62.31**	**3.56**
ResDSN [48]	SegBaseline	89.89	1.12	71.10	6.25	55.91	4.24
	DDT	92.57	1.10	**76.60**	5.03	59.29	4.19

4.3 Results on MSD chanllenge:

Methods	Average DSC (%)
DDT (Ours)	**63.43**
nnU-Net [19]	63.00
UMCT [43]	63.00
K.A.V.athlon	62.00
LS Wang's Group	55.00
MIMI	60.00
MPUnet [26]	59.00