DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Biomedical Computing

Skeleton Prediction of 3D Biomedical Structures

Alok Verma

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Biomedical Computing

Skeleton Prediction of 3D Biomedical Structures

Skelettprognose von 3D Biomedizinisch Strukturen

Author:	Alok Verma
Supervisor:	Prof. Dr. Laura Leal-Taixé
Advisor:	Prof. Dr. Hanspeter Pfister and Dr. Donglai Wei
Submission Date:	April 15, 2020

I confirm that this master's thesis in biomedical computing is my own work and I have documented all sources and material used.

Munich, April 15, 2020 Alok Verma

Acknowledgments

I would like to thank Prof. Hanspeter and Donglai at Harvard for hosting and guiding me, Prof. Laura at TU Munich for accepting my thesis proposal and being so supportive throughout it.

Abstract

3D skeletons are crucial for biomedical image analysis, from neuron tracing to blood vessel centerline prediction. However, due to significant variations in appearance and morphology of objects of interest, there are few frameworks to predict 3D instance skeletons robustly for different imaging domains.

In this thesis, we propose an end-to-end trainable method called *Flux-and-Track* to tackle this challenge. Our method first extends a flux-based semantic skeleton prediction network for 3D data to obtain semantic skeleton prediction. Then, we use an instance proposal module to generate over-split skeleton of all object instances that are later agglomerated by a recurrent tracking network.

We quantitatively evaluate our method on two kinds of benchmark datasets: neural skeletons from electron microscopy and synthetic blood vessels, and qualitatively evaluate on a third brain MRA dataset. We show that our method consistently achieves better results.

Contents

1. Introduction

1.1. Motivation

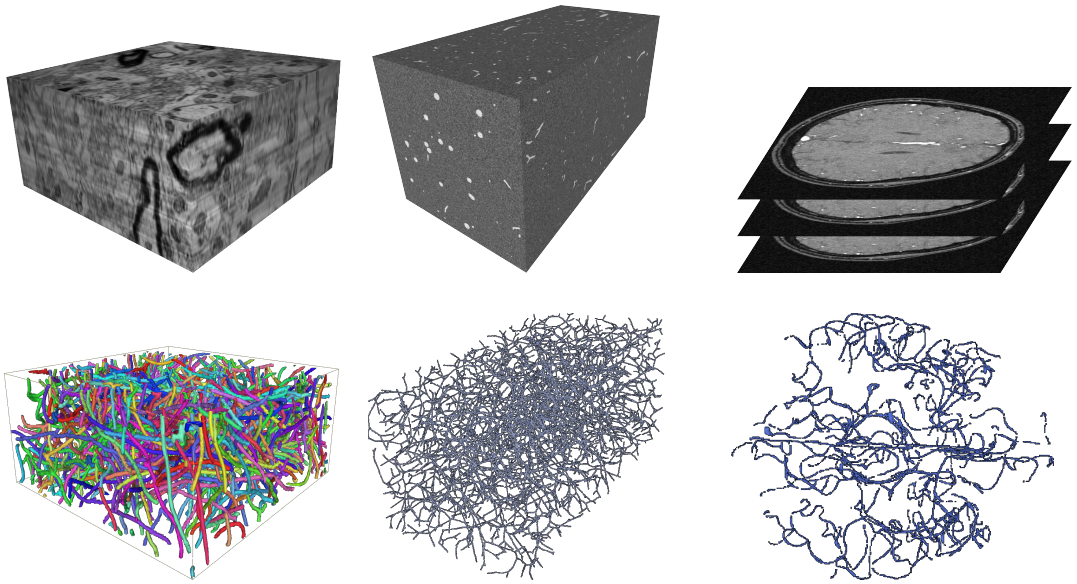
Instance skeleton extraction from volumetric data has numerous applications in the biomedical domain. For example, for neural circuit analysis [35] accurate wiring diagrams with skeletons and their synaptic connections can enable new insights into the workings of the brain and advance bio-inspired artificial intelligence. For clinical blood vessel analysis from CT images [26] we need accurate centerline and bifurcation prediction to quantify structural or flow patterns.

Another important application area is Connectomics [28, 7, 15] where although segmentation of electron microscopy(EM) images is a major step but it is not the final goal. It is important to find out interconnections between neurons, identify common neurons shapes [34], find geodesic distance between synapse connections and soma etc. Such analysis naturally calls in for skeletonization of neurons. A straight forward way is to skeletonize the predicted segmentations for which many algorithms already exist [23, 20] but it is hard to obtain segmentation ground truth and state-of-the-art methods are either slow or produce results with many false merges and false splits.

Apart from serving as a connectome analysis tool, skeletons could be instrumental in improving the segmentations itself. Since, they capture a global topology of segments they can be a indicator of false split and false merges in segmentation. A recent method by Matejek *et al.* [18] utilized skeleton end-points to identify false splits and merged them using skeleton curvature information.

Creating ground truth skeleton data is also easier than segmentation, dedicated tools like KNOSSOS [13] exist for easy labelling. Berning *et al.* [3] quotes skeleton labelling to be 25 to 100 times faster than segmentation labelling. It also proposes a semi-automatic segmentation method based on hand traced neurons. In another kind of neural images obtained from fluorescence confocal microscopy, neuron tracing methods [12] are being developed which is essentially skeletonization.

However, instance skeleton extraction is a challenging task. Due to complex 3D geometry. There can be false split and false merge errors among branches even for one instance, e.g. vascular tree. Besides, in brain electron microscopy (EM) images neurons are densely packed and their appearances have diverse textures which adds further complexity to the neural skeletonization process.



(a) (b) (c)

Figure 1.1.: 3D Instance Skeleton Prediction. We predict (a) neuron skeletons from EM images, (b) blood vessel centerline from CT, and (c) brain vessel centerline from MRA

There are two common approaches for instance skeleton extraction. One approach computes instance segmentation and applies thinning methods to obtain instance skeletons [10]. The other approach computes semantic skeletons then uses connected components to distinguish different instances [31]. In the first approach, obtaining satisfactory annotations is costly, it is much simpler to trace skeletons than delineate the detailed boundaries for object segmentation, *e.g.*, neuron tracing in EM images. The second approach is prone to false merge errors when the skeletons are close to each other and false split errors when ambiguity exists in the image appearance.

In this work, we propose an end-to-end pipeline extracting instance-level skeletons for tubular structures. The pipeline includes three steps.

- First, we extend a 2D flux-based method [30] to predict 3D semantic skeletons.
- Second, we create over-split skeleton instances.
- In the last step, we connect the over-split instances using a recurrent tracking network.

We conduct quantitative evaluation on two public datasets from different biological domains (Figure 1.1 (a, b)) and qualitative evaluation on a third dataset (Figure 1.1 (c)).

We show that the method learns high-quality intermediate representations for skeleton prediction. It also effectively overcomes the problem of split and merge errors suffered by most current end-to-end solutions. We compare our method with state-of-the-art methods [5, 29, 30].

1.2. Related Work

We review relevant skeleton extraction methods in 2D and 3D. We also review few instance segmentation methods for 3D EM images as it is similar to instance skeletonization problem.

1.2.1. Skeleton Extraction

Earlier learning-based methods were developed by formulating skeleton extraction into a regression problem [1] or leveraging the hypothesis that the contexts around skeletons are symmetric [27]. Current state-of-the-art methods are mostly deep learning based which can be classified into two categories: 1) direct binary pixel classification [26, 33, 17]; 2) encoding skeletons in an intermediate representation [29, 30, 24]. For the first approach, DeepVessel [26] proposed cross-hair filters from three intersecting 2-D filters to reduce training overhead while preserving 3D context information. Hi-Fi [33] proposes a new CNN architecture leveraging multi-scale features and bidirectional guidance to make binary predictions. In the second category, skeletons are encoded into intermediate representations like distance transform and context flux. The method in [29] generates pseudo skeletons from learned nearest distance from any point to the tubular structure surface. DeepFlux [30] operates on natural images and learns relationships between image pixels and their closest skeletal points by flux field and recovers skeletons using magnitude and directions of predicted flux. It yields the best performance and therefore, we modify and extend DeepFlux and train our network to learn 3D flux as intermediate representation for skeletons.

1.2.2. Recurrent Methods for Instance Prediction

We review few state-of-the-art methods which predicts curvilinear roads or its boundaries. This is in many ways similar to instance skeletonization.

Recently proposed method by Liang *et al.* [16] predicts road boundaries using polyline representations by using a convolutional recurrent network, predicting one vertex at a time. An extension of their work, DAG-Mapper [8], takes into account road topology and predicts splits and merges too. Another recurrent road extraction method,

RoadTracer [2], predicts road topology using a iterative search method guided by a convolutional neural network.

Another related work, instance segmentation is also a similar problem like instance skeletonization, and a well known recurrent segmentation method used in the field of Connectomics [28, 7, 15] is flood-filling network (FFN) [10]. It performs instance segmentation on 3D EM images by starting from a seed position and iteratively making predictions for overlapping sliding windows. Despite their outstanding performance, training a recurrent model is time-consuming. To overcome the above problem, instead of growing the entire structure step by step iteratively, we only apply the recurrent method on sparse locations which results in a more efficient algorithm.

1.2.3. Error detection and Error Correction

Instance skeletonization problem is similar to instance segmentation in a way that both can have false merges and splits. Therefore we can adopt ideas from segmentation methods in skeletonization to solve false merges and splits. In Connectomics [28, 7, 15], most connectomes reconstruction pipelines include an instance-level automatic error correction step. Zung *et al.* [36] trains classifiers to detect false splits and false merges for each object. They first train an error detection network to output an error map with the same size as the input image patch, centered at an voxel inside the object of interest. The second stage prunes object mask containing only merge errors obtained from superset of baseline object mask and detected error areas. However, the success of error correction largely depends on the accuracy of the error detection net. Another method by Matejek *et al.* [18] extract graphs from initial segmentations and exploit specific geometric and topological properties from underlying biological morphology to solve merge errors.

2. Method

We propose a novel method for instance skeletonization in 3D biomedical images called Flux-and-Track. The method is partly motivated by DeepFlux [30] but differs in following ways:

- It separates skeletons for each individual segment - called as instance skeletonization.
- It is applied on 3D biomedical images instead of 2D images.
- After initial prediction, a splitting and matching step based on skeleton topology is devised to remove false merges and false splits.

Our Flux-and-Track method consists of three stages: 3D flux prediction using a flux network, instance extraction and over-split skeleton generation, and finally skeleton agglomeration by merging split skeletons. Figure 2.1 depicts the overall pipeline and following sections explains the steps in details.

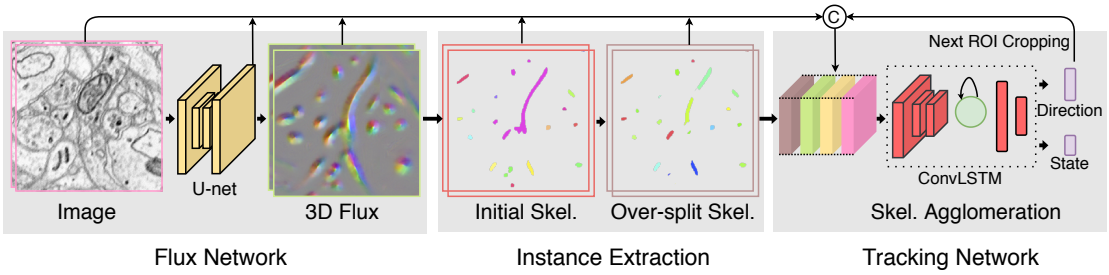
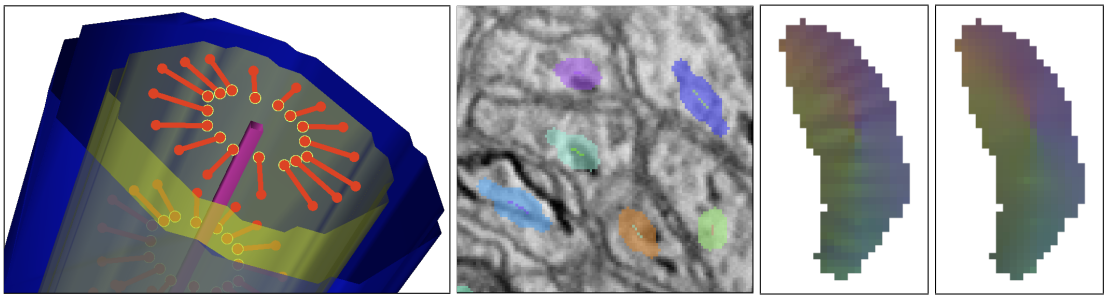


Figure 2.1.: Overview of our Flux-and-Track method. The flux network takes raw image as input and outputs flux predictions. Next, the predicted flux goes through the instance extraction step to generate over-splitted instance proposals. Finally, we agglomerate the proposals based on predictions from the tracking network, namely the growing direction and *continue-stop* state.



(a) (b) (c) (d)

Figure 2.2.: **3D skeleton context flux.** (a) Blue: a synthetic object segment, Purple: the skeleton; Green: context region of the skeleton; Red: flux vectors pointing away from the skeleton, (b) Neural skeletons and their context regions, (c) Color-coded flux field. Roughness due to discrete skeleton points (d) Smooth flux field after interpolating skeletons using splines. (Best viewed in electronic version).

2.1. Flux Network for 3D Flux Prediction

2.1.1. Flux Definition

We formulate skeleton prediction as a regression problem and define a 3D flux field from ground truth skeleton points Ω_s following the 2D flux representation in [30]. Centered at each skeleton point, the *context region* Ω_c is defined as the set of points inside 3D balls of radius r centered at the skeleton points:

$$\Omega_c = \{\mathbf{x} : \min_{\mathbf{y}\in\Omega_s} \|\mathbf{x}-\mathbf{y}\|_2 < r\} \tag{2.1}$$

Further a scalar distance function D is defined from the skeleton points inside the domain Ω as follows:

$$\mathbf{D}(\mathbf{x}) := \begin{cases} \min_{\mathbf{y}\in\Omega_s} \|\mathbf{x}-\mathbf{y}\|_2 & \text{if } \mathbf{x}\in\Omega_c \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Finally, flux $\overline{D}$, which is a $\mathbb{R}^3$ vector field is obtained by taking discrete gradient of D:

$$\overline{\mathbf{D}}(\mathbf{x}) := \begin{cases} \nabla\mathbf{D} & \text{if } \mathbf{x}\in \text{int } \Omega_c \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

In essence $\overline{\mathbf{D}}$ is a field with non-zero direction vectors defined in the *context region* of skeletons, such that the vectors are pointing away from the skeletons, as shown in Figure 2.2 (a).

But such flux is non-smooth if the skeletons are defined on a discrete grid, shown in Figure 2.2 (c). Learning such a non-smooth field is not encouraged as deep convolutional networks usually fail to generate such sharp fields. Hence, to create smoother field (Figure 2.2 (d)), skeleton points are interpolated using splines and distance transform is computed using the interpolated points.

The advantages for encoding skeletons in such a field are:

- Flux Network has to learn to look for both global and local properties while predicting the field. This helps to avoid local false merges
- Voxel wise loss function for the deep net can be easily constructed and the training objective can be agnostic of the number of skeleton instances.
- Predicted field can be useful to solve false merges and splits in later post processing steps.

2.1.2. Network Design

Since flux field representation is non-local and highly dependent on underlying 3D shapes we use a fully convolutional 3D U-net to predict the 3D flux field from 3D images to enable versatile representations from the training objects. Details about the network architecture are shown in the appendix (Figure A.2).

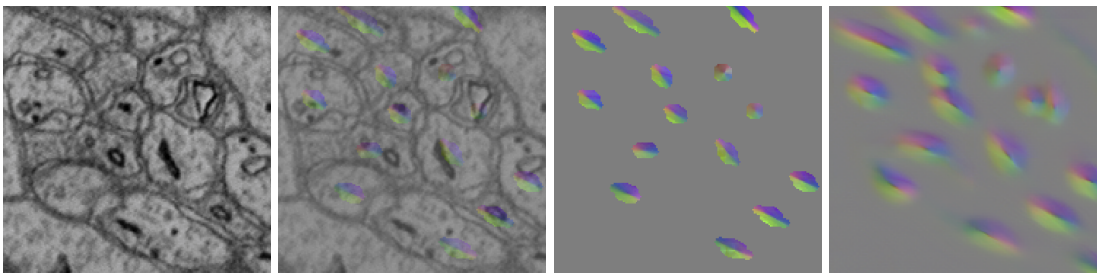
2.1.3. Training Objective

$$\mathcal{L}_{cos}(\mathbf{P}, \mathbf{T}) := \sum_{\mathbf{x} \in \Omega} \mathbf{W}(\mathbf{x}) \left(1 - \frac{\mathbf{P}(\mathbf{x}).\mathbf{T}(\mathbf{x})}{\max(\|\mathbf{P}(\mathbf{x})\|_2.\|\mathbf{T}(\mathbf{x})\|_2, \epsilon)}\right) \tag{2.4}$$

$$\mathcal{L}_{mse}(\mathbf{P}, \mathbf{T}) := \sum_{\mathbf{x} \in \Omega} \mathbf{W}(\mathbf{x}) \| \|\mathbf{P}(\mathbf{x})\|_2 - \|\mathbf{T}(\mathbf{x})\|_2 \|_2^2 \tag{2.5}$$

$$\mathcal{L}_{flux} := \alpha \mathcal{L}_{cos} + (1 - \alpha) \mathcal{L}_{mse} \tag{2.6}$$

Though a L2 or L1 loss between target field $\mathbf{T}$ and predicted field $\mathbf{P}$ can be used, but a more stringent loss would be to enforce correct prediction of the directions of the vectors, hence weighted cosine similarity is used to calculate the loss (Equation 2.4). This forces a sharper prediction and ensures that the network directly learns directions and hence the shape of the neurons. Weight $\mathbf{W}$ balances the loss contribution from context and non-context regions, defined as $|\overline{\Omega_c}|/|\Omega_c|$ for context region and 1 otherwise. Since, outside skeleton context region the vector field has zero magnitude and therefore cosine loss is meaningless, a weighted combination (Equation 2.6) of mean square error (Equation 2.5) and cosine loss (Equation 2.4) is used. Figure 2.3 shows the learned flux prediction.



(a) (b) (c) (d)

Figure 2.3.: (a) and (c) shows an 2D slice of input data and the ground truth field. They are overlayed in (b). (d) shows the predicted field.

2.2. Instance Extraction for Over-split Skeletons

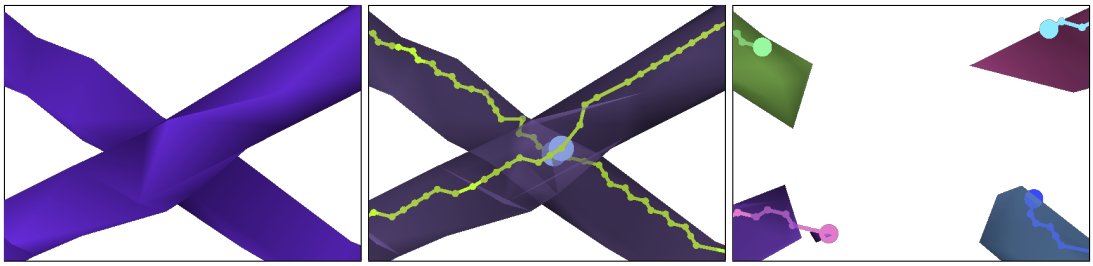
To obtain the instance skeletons back from such an encoding, a simple postprocessing step is devised. First observation about the predicted field is: in the vicinity of skeleton voxels they point away from each other, while at non skeleton voxels they point almost in the same direction. This property can be used to identify skeleton voxels, *divergence* at skeleton points would be high, where as for all other locations it would be low. Thus, thresholding the *divergence* would allow to create a skeleton mask. For separating skeletons of different instances connected components analysis can be performed.

2.2.1. Topological Skeleton Graph

We thin the predicted skeletons to a single voxel width and create an undirected graph, shown in Figure 2.4 (b). Based on the degree of incident edges, we can identify the vertices V into junctions $J := \{n \in V : \text{degree}(n) > 2\}$ and endpoints $E := \{n \in V : \text{degree}(n) = 1\}$.

2.2.2. Generating Over-split Skeletons

While connected component analysis separates skeleton instances of most segments, but closely located skeletons crossing each other could be falsely merged. This would create junctions unless skeletons of two parallel segments are merged throughout which is unlikely. So we utilize the topological skeleton graph to split skeletons at junctions J (Figure 2.4). Predicted skeletons are split by partitioning with a plane as orthogonal as possible to the merged skeletons. Such a plane can be defined using Singular Value Decomposition of all skeletons points near the junction.



(a) (b) (c)

Figure 2.4.: **Over-split skeletons.** (a) Thick and falsely merged predicted skeletons (b) Topological graph constructed by thinning and identified junction point (c) Result of splitting skeletons at junction point

This process creates over-split skeletons (Figure 2.4 (c)) which can be agglomerated using another post-processing step.

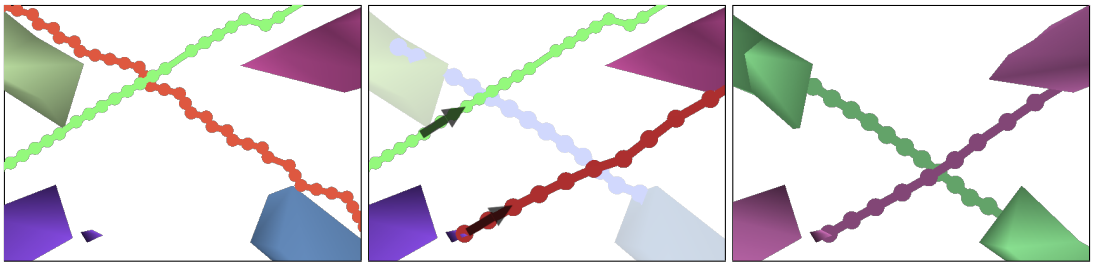
2.3. Tracking Network for Skeleton Agglomeration

2.3.1. Network Design

To iteratively grow and merge skeletons from endpoints, we train a 3D Conv-LSTM-FC based recurrent tracking network (Figure 2.1). While growing, if the predicted trajectory hits another skeleton, both skeletons are merged together. Every growing step produces two outputs: first, the direction along which the skeleton is to be grown, which is scaled by a constant λ to jump to the next position; second, a binary state variable *{continue-stop}* controlling if the growing should be terminated at current position. The input to the network is a small ROI from previous layers, initially centered at a skeleton end and moved as the tracking proceeds. The layers consist of the image, skeleton mask and Flux Net features which provides a larger spatial context to the Tracking Network. Details about the network architecture is shown in the appendix (Figure A.3).

2.3.2. Loss Functions

We use the predicted over-split skeletons from the Flux Network as seeds and ground truth skeleton graph G_{gt} to find a path between pairs of over-split skeleton segments. To minimize the influence from artifacts stemming from all the stages, including annotation, thinning and smoothing (Figure 2.5 (a)), ground truth skeletons only provides guidance for tracking. Starting from an end point, an oracle can track by only utilizing directions from nearest nodes in the ground truth skeleton graph, shown



(a) (b) (c)

Figure 2.5.: Skeleton agglomeration. (a) Misaligned predicted and ground truth skeletons. (b) Oracle generated tracking paths using ground truth skeleton directions as guidance (c) Over-split skeleton pairs correctly agglomerated by Tracking Network.

in Figure 2.5 (b). These on-the-fly directions supervise the learning of our Tracking Network.

We want our predictions to follow the ground truth (GT) path smoothly and loosely without running astray. Therefore, at every step, we dynamically calculate the target direction $\mathbf{u_t}$ using the tangential direction from multiple closest GT nodes and also the normal direction to the GT path. The target state s_t is *continue* until the model tracks till the corresponding split skeleton pair or reaches the GT graph end. We define loss functions for directions and path states as follows:

$$\mathcal{L}_{\text{dir},\tilde{t}} = \sum_{t=\tilde{t}}^{\tilde{t}+\frac{N}{2}} \left(1 - \frac{\mathbf{u_t}.\mathbf{v_t}}{\max(\|\mathbf{u_t}\|_2.\|\mathbf{v_t}\|_2,\, \epsilon)}\right), \tag{2.7}$$

$$\mathcal{L}_{\text{state},\tilde{t}} = \sum_{t=\tilde{t}}^{\tilde{t}+\frac{N}{2}} w_t \left(-s_t \log(r_t) - (1-s_t)\log(1-r_t)\right). \tag{2.8}$$

$$\mathcal{L}_{\text{track},\tilde{t}} = \mathcal{L}_{dir,\tilde{t}} + \beta \mathcal{L}_{state,\tilde{t}} \tag{2.9}$$

$$w_t = \begin{cases} 1 & \text{if } s_t = continue \\ t & \text{if } s_t = stop \end{cases} \tag{2.10}$$

We use cosine similarity (Equation 2.7) as a training loss for directions and weighted binary cross entropy (Equation 2.8) for path states. To avoid the tracking going astray during initial epochs, we perform multiple-phase training with increasing max steps of $N = \{4, 8, 16, 32\}$ and optimize the tracking loss every $\frac{N}{2}$ steps.

2.4. End-to-End Fine Tuning

Initially, we train the flux network and the tracking network separately for the dependency relationship between two networks. We perform final fine-tuning end-to-end by jointly optimizing linear combination of all losses as shown in Equation 2.11.

$$\mathcal{L} = \alpha\mathcal{L}_{cos} + \beta\mathcal{L}_{scale} + \gamma\mathcal{L}_{dir,\tilde{t}} + \phi\mathcal{L}_{state,\tilde{t}}. \tag{2.11}$$

The scaling constants $\alpha, \beta, \gamma, \phi$ are determined empirically and satisfy the constraint: $\alpha + \beta + \gamma + \phi = 1$.

2.5. Data Augmentation

Augmenting 3D data while training has been very effective in improving affinity prediction for EM segmentation [32, 15]. Borrowing the training set augmentations from previous methods [15, 6], the following augmentations were applied for Flux Network.

- **Rotations** of 90° degree and also incremental along Z axis.
- **Flips** in x, y and z dimensions.
- **Rescaling** images, skeleton and subsequently the field which was done prior to training.
- **Gaussian blur** of a random set of $2D$ slices in input volume. This mimics the actual scenario when only a few slices are out-of-focus and intermittently spread in $3D$.
- **Elastic deformation** of images. A random perturbing field is defined in such a way that the images are only slightly deformed with a max of 6 pixels, which would not significantly move skeletons. Hence, this can be done on-the-fly during training. This augmentation can force the network to look for 3D shape features rather than focusing entirely on boundaries.
- Random **brightness**, **contrast** adjustments and **gamma-correction**.
- **Missing parts**, where a random part of the $2D$ slice is replaced with a flat grayscale value. This mimics artifacts in $2D$ slices due to physical folding, knife marks etc.

- **Missing section**, where few $2D$ slices are removed from the input stack. This mimics the case when few slices are lost and not imaged due to other issues. This causes a discernible discontinuity along Z axis.
- **Misalignment**, where a few randomly chosen slices are displaced laterally by few pixels. Which can occur in a real scenario when alignment is not perfect due to image artifacts.

For Tracking Network only 90° Rotations, Flip and Transpose augmentation operations were used.

3. Experiments

3.1. Datasets

We train and test proposed method on three datasets. On the quantitative side, we use a synthetic CT vessel dataset and a real EM dataset. For qualitative comparison we use an MRA dataset of brain vessels.

- **Synthetic CT Vessel ([26]).** We use synthetic CT vessel dataset (Figure 1.1 (b))for centerline detection task. The synthetic dataset consists of isotropic single-instance volumes of size $600 \times 304 \times 325$. We use 20 volumes for training and 15 for testing.
- **SNEMI3D ([25]).** For neuron skeleton extraction, we conduct experiments on SNEMI3D (Figure 1.1 (a)), a neuron segmentation benchmark for EM images containing packed and complex-shaped neuron instances. We create ground truth skeletons from segmentation by topological thinning [21]. The train volume is of size $100 \times 1024 \times 1024$ voxels with $30 \times 6 \times 6$ nm resolution. As the test label is unavailable, we report results on a validation volume from the original paper [11].
- **Brain MRA ([4]).** Brain MRA dataset (Figure 1.1 (c)) consists of 42 volumes of annotated blood vessels of size $128 \times 448 \times 448$ and resolution $0.8 \times 0.5 \times 0.5$ mm. We use 28 volumes for training and show result on one test volume. We only show qualitative results because the ground truth skeletons were disconnected and therefore ground truth topology graph could not be created. Also, we found annotations missing for faintly identifiable vessels.

3.2. Evaluation Metric

To evaluate multiple skeleton instances and their connectivity, we choose the *instance-level F1 score* , similar to [16]. Each proposed skeleton is assigned to a single ground truth skeleton based on maximum overlap. All predicted skeleton voxels within a threshold distance to the mapped gt skeleton voxels are classified as true positives and rest as false positives. Ground truth voxels outside the range of predicted voxels are

	Thresholding Value	
	SNEMI	Synthetic CT
UNET-3D [5]	0.90	0.85
DT [18]	0.45	0.45
DeepVesselNet [16]	-	0.30
DeepFlux† [19]	0.15	0.65
Flux-and-Track (**Ours**)	0.55	0.80

Table 3.1.: Table shows the threshold values used to binarize the prediction of different methods. We skip dilation for our Flux-and-Track method.

classified as false negatives. We also report mean *connectivity score (C)*, defined for each ground truth skeleton as $C_i = \frac{1(N_i)}{N_i}$, where N_i is the number of assigned predicted skeletons to ground truth i and $1(\cdot)$ is the indicator function, similar to [16]. We also provide the *pixel-level F1 score* for semantic skeleton prediction as a sanity check.

3.3. Methods in Comparison

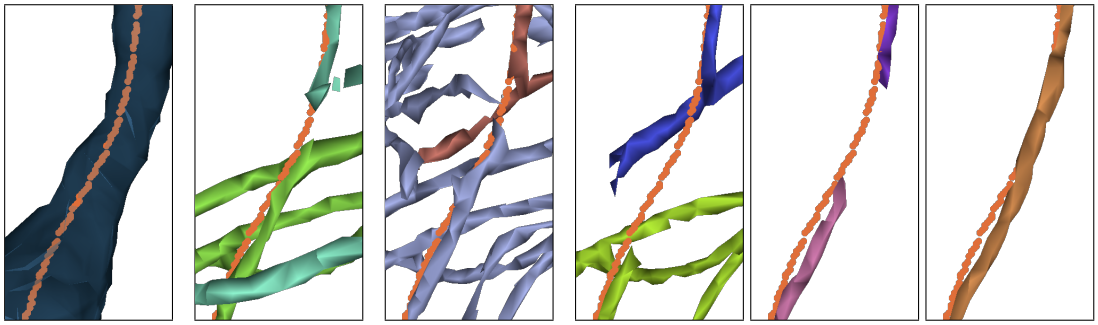
We compare our method with state-of-the-art deep learning based skeletonization methods defined as follows:

- Predict *dilated binary mask* [5, 26].
- Predict *inverse truncated distance transform of skeletons* [29].
- *DeepFlux*† [30] extended from 2D skeleton cases.
- *Centerline Prediction* [24] (only for qualitative comparison on the MRA, results obtained from [14]).

We use the same U-net architecture and hyper-parameters for all methods. For each method we find the best thresholding parameter using grid search as shown in Table 3.1. We also dilate and erode the binarized predictions with kernel size of 3 and 4 respectively for all methods as done in the DeepFlux paper [30]. We skip dilation for our Flux-and-Track method.

	SNEMI			Synthetic CT
	F1-pix.	F1-ins.	C	F1-pix./F1-ins.
UNET-3D [5]	**0.924**	0.695	0.625	0.833
DT [29]	0.908	0.655	0.559	**0.983**
DeepVesselNet [26]	-	-	-	0.779
DeepFlux† [30]	0.792	0.741	0.500	0.950
Flux-only (**Ours**)	0.773	0.716	0.730	0.926
Flux-and-Track (**Ours**)	0.810	**0.764**	**0.733**	-

Table 3.2.: Comparison on SNEMI and synthetic vessel dataset. DeepFlux†: extension of [30] to predict 3D skeletons instead of 2D.



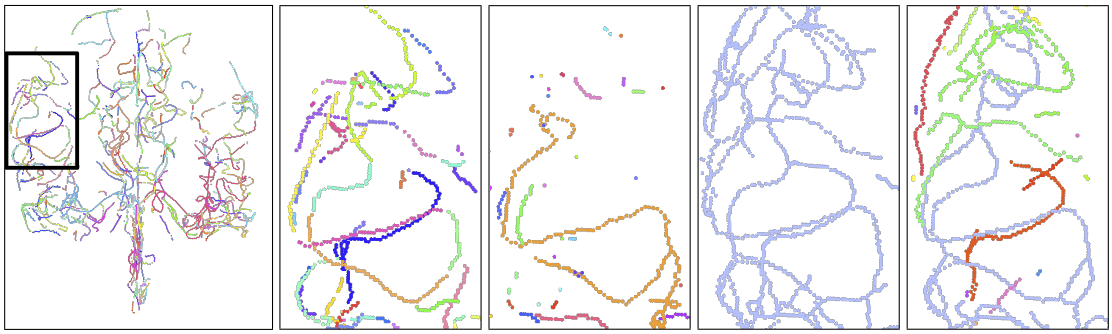
(a) (b) (c) (d) (e) (f)

Figure 3.1.: Qualitative results on SNEMI. (a) Ground Truth skeleton and segment (b) UNET-3D [5] (c) DT [29] (d) DeepFlux† [30] (e) Our result after splitting (f) Our result after tracking

3.4. Results

3.4.1. Quantitative

As shown in (Table 3.2) our method outperforms other approaches for real dataset in *instance-level F1* and *connectivity* scores signifying that our method produces less instance false splits and merges. Other methods perform better on synthetic dataset because of its single instance and simple shapes. We do not run tracking on synthetic dataset as it has only a single instance, connectivity score is also not calculated therefore. We show SNEMI3D qualitative results in Figure 3.1. As can be noticed all other methods have severe false splits and merges, while our method performs better by learning more robust intermediate representations and utilizing topology information to resolve errors.



(a) (b) (c) (d) (e)

Figure 3.2.: Qualitative MRA Results. (a) GT (b) Zoomed region from GT (c) Result from [24] (d) Result from [5] showing a false merge between disconnected vessels (e) Our result demarcates close vessels successfully.

3.4.2. Qualitative

We also visualized (Figure 3.2) our results on the Brain MRA dataset[4], which has 42 volumes of annotated blood vessels of size $128 \times 448 \times 448$ and resolution $0.8 \times 0.5 \times 0.5$ mm. We use 28 volumes for training and show result of one test volume. We only show qualitative results because the ground truth skeletons were disconnected and also missing for many faint vessels. We compare our method with three other state-of-the-art methods in which Sironi *et al.* [24] doesn't manage to detect as many vessel instances as ours and a 3D Unet predicting binary output (UNET-3D) [5]. Although UNET-3D [5] can also perform effective detection in regions with artifacts, it fails to predict correct topology as shown in Figure 3.2 (d).

4. Future Work

Our proposed method shows promising results for skeleton prediction in complicated 3D datasets but it still has unresolved false splits because the Tracking Network fails to correctly track in longer gaps. Also our splitting method creates more than necessary splits which could be avoided. So further work would be to:

- Improve splitting process by selectively splitting at false splits. Currently all junction points are split but it is not necessary as some of them are branch points in the same object. A classifier can be designed to discriminate between false merges and branch points.
- Improve the training ground truth for Tracking Network. If there is a large mismatch between ground truth skeletons and predicted over-split skeletons the Tracking network may not learn meaningful filters. To alleviate this, synthetic Tracking Network ground truth can be constructed.

Also, one could find better error metrics for evaluating instance skeletonization performance since binary pixel level F1 score does not care about instances and even instance F1 scores are biased towards split skeletons. One potential candidate would be estimated run length metric from Januszewski *et al.* [9, 10]. Last, as an application for skeletons, skeleton-assisted segmentation methods can be devised which could alleviate false merges and splits in segmentation methods and also use cheaper skeleton labels for training.

5. Conclusion

We developed an instance skeletonization method for 3D biomedical images and showed its efficacy on different modalities and complicated shapes. We demonstrated our novel usage of topology of skeletons to resolve false merges which resulted in great improvement. While our method beats other state-of-the-art skeletonization approaches it still results in over-split skeletons. This means the splitting and tracking steps needs to be optimized further to achieve perfect results.

A. Appendix

A.1. Network Architectures

A.1.1. Flux Network

Input to the network is a grayscale image and output is 3 channel vector field. The backbone of the network is based on Unet [22]. The encoder and decoder of the Unet has four stages, and a center bottleneck layer, with $8, 16, 24, 32, 40$ filters respectively. The network has a theoretical field of view of $32 \times 378 \times 378$. Further details can be seen in Figure A.1 and Figure A.2.

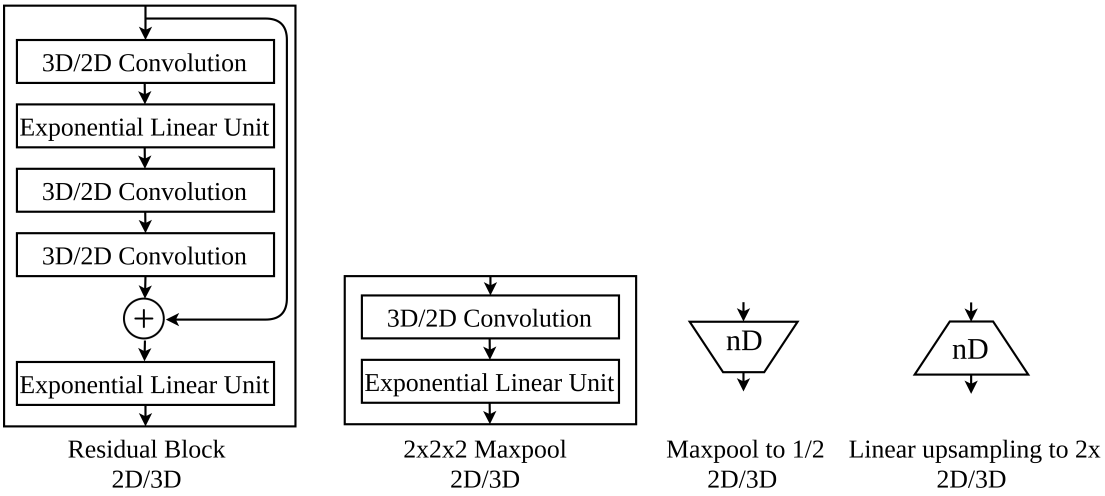


Figure A.1.: Network blocks used in Flux Network and Tracking Network.

A.1.2. Tracking Network

Tracking Network is based on 3D convolution layers, a convolutional LSTM block and fully connected layers at the end. The details are show in Figure A.3.

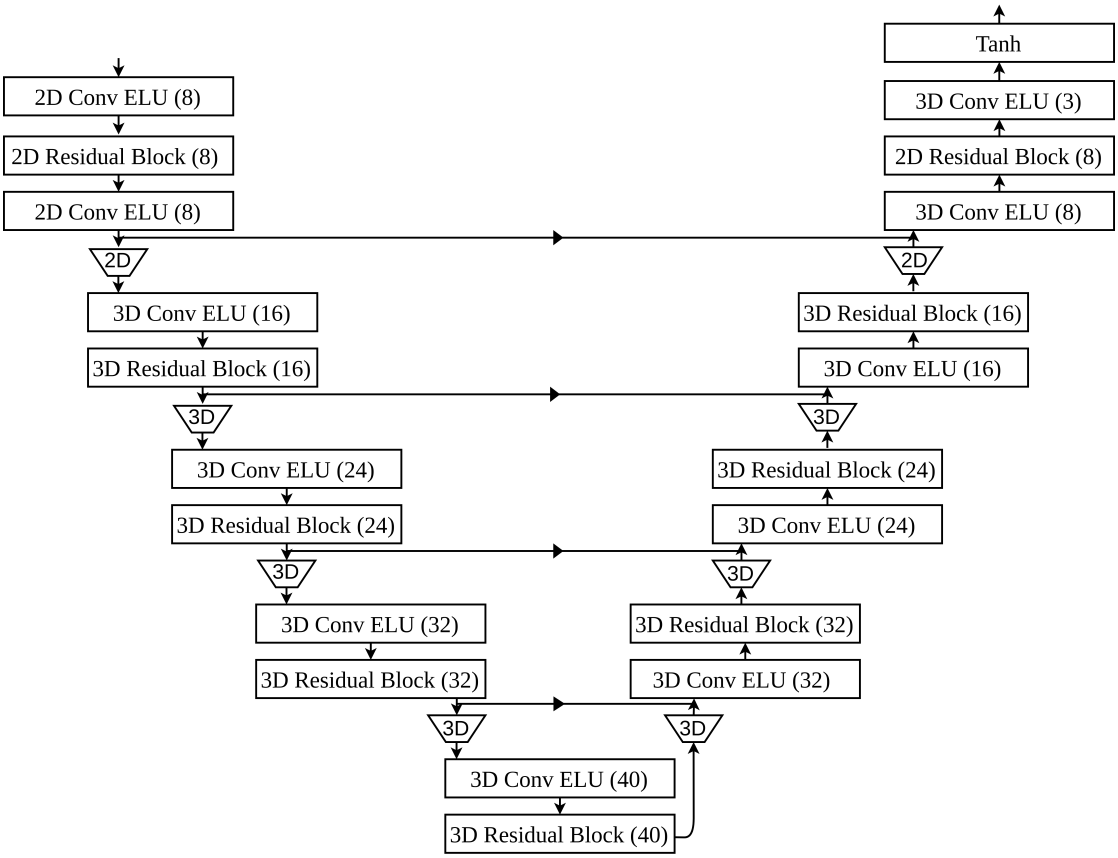


Figure A.2.: Flux Network. Input is a grayscale image of size $1 \times 64 \times 192 \times 192$. Output is the flux field of size $3 \times 64 \times 192 \times 192$.

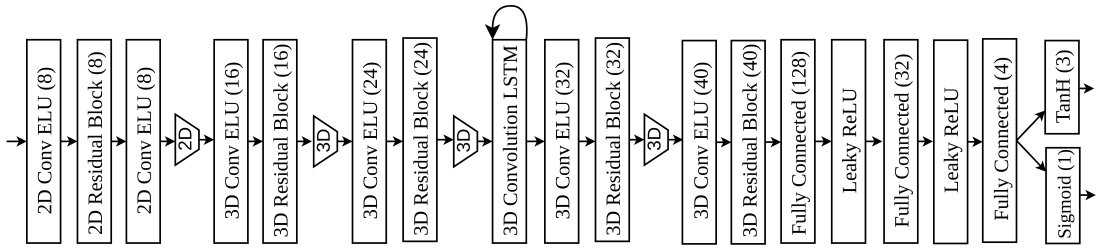


Figure A.3.: Tracking Network. Input is a $14 \times 16 \times 64 \times 64$ volume, comprising of input image, flux, start skeleton mask, other skeletons mask, and global features from Flux Network of channel sizes $1, 3, 1, 1, 8$ respectively. Outputs are 1) growing direction vector of size 3 and a scalar for the *{continue-stop} probability.*

List of Figures

List of Tables

Bibliography

[1] V. L. Amos Sironi and P. Fua. "Multiscale centerline detection by learning a scale-space distance transform." In: *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[2] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt. "RoadTracer: Automatic Extraction of Road Networks from Aerial Images." In: *Proceedings of the 2018 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[3] M. Berning, K. Boergens, and M. Helmstaedter. "SegEM: Efficient Image Analysis for High-Resolution Connectomics." In: *Neuron* 87.6 (Sept. 2015), pp. 1193–1206.

[4] E. Bullitt, D. Zeng, G. Gerig, S. Aylward, S. Joshi, J. K. Smith, W. Lin, and M. G. Ewend. "Vessel Tortuosity and Brain Tumor Malignancy." In: *Academic Radiology* 12.10 (Oct. 2005), pp. 1232–1240. DOI: 10.1016/j.acra.2005.05.027.

[5] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. "3D U-Net: learning dense volumetric segmentation from sparse annotation." In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2016, pp. 424–432.

[6] ELEKTRONN = Deep learning toolkit for high throughput analysis of large scale 2D and 3D images with convolutional neural networks. URL: http://elektronn.org/.

[7] J. Funke, F. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, and S. C. Turaga. "Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome Reconstruction." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7 (July 2019), pp. 1669–1680. DOI: 10.1109/tpami.2018.2835450.

[8] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun. "DAGMapper: Learning to Map by Discovering Lane Topology." In: *The IEEE International Conference on Computer Vision (ICCV)*. 2019.

[9] M. Januszewski, J. Kornfeld, P. H. Li, A. Pope, T. Blakely, L. Lindsey, J. Maitin-Shepard, M. Tyka, W. Denk, and V. Jain. "High-Precision Automated Reconstruction of Neurons with Flood-filling Networks." In: (Oct. 2017). DOI: 10.1101/200675.

[10] M. Januszewski, J. Kornfeld, P. H. Li, A. Pope, T. Blakely, L. Lindsey, J. Maitin-Shepard, M. Tyka, W. Denk, and V. Jain. "High-precision automated reconstruction of neurons with flood-filling networks." In: *Nature Methods* (2018).

[11] N. Kasthuri, K. Hayworth, D. Berger, R. Schalek, J. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, T. Jones, M. Roberts, J. Morgan, J. Tapia, H. S. Seung, W. Roncal, J. Vogelstein, R. Burns, D. Sussman, C. Priebe, H. Pfister, and J. Lichtman. "Saturated Reconstruction of a Volume of Neocortex." In: *Cell* 162.3 (July 2015), pp. 648–661. DOI: 10.1016/j.cell.2015.06.054.

[12] C. Kayasandik, P. Negi, F. Laezza, M. Papadakis, and D. Labate. "Automated sorting of neuronal trees in fluorescent images of neuronal networks using NeuroTreeTracer." In: *Scientific Reports* 8.1 (Apr. 2018).

[13] KNOSSOS - 3D Image Visualization and Annotation Tool. URL: https://knossos.app (visited on 11/30/2019).

[14] M. Koziński, A. Mosinska, M. Salzmann, and P. Fua. "Learning to Segment 3D Linear Structures Using Only 2D Annotations." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 283–291.

[15] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung. *Superhuman Accuracy on the SNEMI3D Connectomics Challenge*. 2017. arXiv: 1706.00120 [cs.CV].

[16] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun. "Convolutional Recurrent Network for Road Boundary Extraction." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[17] X. Liu, P. Lyu, X. Bai, and M. Cheng. "Fusing Image and Segmentation Cues for Skeleton Extraction in the Wild." In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Oct. 2017, pp. 1744–1748. DOI: 10.1109/ICCVW.2017.205.

[18] B. Matejek, D. Haehn, H. Zhu, D. Wei, T. Parag, and H. Pfister. "Biologically-Constrained Graphs for Global Connectomics Reconstruction." In: *CVPR*. 2019.

[19] B. Matejek, D. Wei, X. Wang, J. Zhao, K. Palágyi, and H. Pfister. "Synapse-Aware Skeleton Generation for Neural Circuits." In: *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 227–235. DOI: 10.1007/978-3-030-32239-7_26.

[20] K. Palágyi. "A sequential 3D curve-thinning algorithm based on isthmuses." In: *Advances in Visual Computing - 10th International Symposium, ISVC 2014, Proceedings*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, 2014, pp. 406–415.

[21] K. Palágyi. "A sequential 3d curve-thinning algorithm based on isthmuses." In: *International Symposium on Visual Computing*. Springer. 2014, pp. 406–415.

[22] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[23] M. Sato, I. Bitter, M. Bender, A. Kaufman, and M. Nakajima. "TEASAR: tree-structure extraction algorithm for accurate and robust skeletons." In: *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*. IEEE Comput. Soc. DOI: 10.1109/pccga.2000.883951.

[24] A. Sironi, E. Türetken, V. Lepetit, and P. Fua. "Multiscale centerline detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.7 (2015), pp. 1327–1341.

[25] SNEMI EM Segmentation Challenge and Dataset.

[26] G. Tetteh, V. Efremov, N. D. Forkert, M. Schneider, J. Kirschke, B. Weber, C. Zimmer, M. Piraud, and B. H. Menze. "DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes." In: *ArXiv* abs/1803.09340 (2018).

[27] K. I. Tsogkas S. "Learning-Based Symmetry Detection in Natural Images." In: *Computer Vision – ECCV 2012*. 2012.

[28] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung. "Convolutional Networks Can Learn to Generate Affinity Graphs for Image Segmentation." In: *Neural Computation* 22.2 (Feb. 2010), pp. 511–538. DOI: 10.1162/neco.2009.10-08-881.

[29] Y. Wang, X. Wei, F. Liu, J. Chen, Y. Zhou, W. Shen, E. K. Fishman, and A. L. Yuille. "Deep Distance Transform for Tubular Structure Segmentation in CT Scans." In: *arXiv preprint arXiv:1912.03383* (2019).

[30] Y. Wang, Y. Xu, S. Tsogkas, X. Bai, S. Dickinson, and K. Siddiqi. "DeepFlux for Skeletons in the Wild." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[31] W. Xu, G. Parmar, and Z. Tu. "Geometry-Aware End-to-End Skeleton Detection." In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2019.

[32] T. Zeng, B. Wu, and S. Ji. "DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation." In: *Bioinformatics* 33.16 (Mar. 2017), pp. 2555–2562. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx188.

[33] K. Zhao, W. Shen, S. Gao, D. Li, and M. Cheng. "Hi-Fi: Hierarchical Feature Integration for Skeleton Detection." In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by J. Lang. 2018. DOI: 10.24963/ijcai.2018/166.

[34] T. Zhao and S. M. Plaza. *Automatic Neuron Type Identification by Neurite Localization in the Drosophila Medulla*. 2014. arXiv: 1409.1892 [q-bio.NC].

[35] Z. Zheng, J. S. Lauritzen, E. Perlman, C. G. Robinson, M. Nichols, D. Milkie, O. Torrens, J. Price, C. B. Fisher, N. Sharifi, et al. "A complete electron microscopy volume of the brain of adult Drosophila melanogaster." In: *Cell* (2018).

[36] J. Zung, I. Tartavull, and H. S. Seung. "An Error Detection and Correction Framework for Connectomics." In: *CoRR* abs/1708.02599 (2017).