

# Intersection To Overpass: Instance Segmentation On Filamentous Structures With An Orientation-Aware Neural Network And Terminus Pairing Algorithm

Yi Liu    Abhishek Kolagunda    Wayne Treible    Alex Nedo    Jeffrey Caplan

Chandra Kambhamettu  
University of Delaware

18 Amstel Ave, Newark, DE, USA 19716

{yliu, abhi, wtreible, aon, jcaplan, chandrak}@udel.edu

## Abstract

*Filamentous structures play an important role in biological systems. Extracting individual filaments is fundamental for analyzing and quantifying related biological processes. However, segmenting filamentous structures at an instance level is hampered by their complex architecture, uniform appearance, and image quality. In this paper, we introduce an orientation-aware neural network, which contains six orientation-associated branches. Each branch detects filaments with specific range of orientations, thus separating them at junctions, and turning intersections to overpasses. A terminus pairing algorithm is also proposed to regroup filaments from different branches, and achieve individual filaments extraction. We create a synthetic dataset to train our network, and annotate real full resolution microscopy images of microtubules to test our approach. Our experiments have shown that our proposed method outperforms most existing approaches for filaments extraction. We also show that our approach works on other similar structures with a road network dataset.*

## 1. Introduction And Related Works

Filamentous structures are ubiquitous among biological systems, such as vesicular networks of fungal hyphae, blood vessels, microtubules, and actin filaments. In order to understand the mechanism of these structures, it is important to quantify their individual characteristics such as the number of filaments, their length, their curvature, and motion. Therefore, it is necessary to segment each individual filament at the instance level from the interconnected network.

Instance segmentation is a fundamental computer vision task, which aims to automatically delineate different objects in an image. While semantic segmentation requires only to predict a category label of each pixel, segmenting at the instance level needs to distinguish individual objects in-

stances, which is considered to be a more challenging task. In recent years, many works have aimed to improve the instance segmentation task. Most approaches [4, 5, 6, 10, 11] for instance segmentation rely on region proposal or object detection using Convolutional Neural Networks (CNNs). These approaches have shown impressive performance on objects with a well-defined boundary and center such as humans, cars, tables, and so on. However, these methods have severe limitations for segmenting filamentous structures for the following reasons: (1). Individual filaments can span the entire length or width of the image; (2). These thin elongated objects are usually close to objects with the same features; (3). Some filamentous network structures are dense and complex, with different filaments overlapping or intersecting each other. These properties of filamentous structures make it hard for proposing a region of interest for individual filaments or detecting them separately. Another problem for machine learning methods is the shortage of labeled datasets for biological filaments. There are datasets available for instance segmentation on more common objects, such as the COCO dataset [7], but as far as we are aware, there is a lack of datasets for segmenting biological filamentous structures at an instance level.

There have been several attempts for segmenting filamentous structure at instance level based on conventional (i.e. non CNN-based) image processing techniques. Xu *et al.* [14] introduced Stretching Open Active Contours(SOAC) with adaptive stretching forces to trace filaments, and used regulated sequential evolution scheme to avoid overlap among SOACs. To deal with the intersection areas, Xu *et al.* dissected converged SOACs into fragments at junctions, then formed new SOACs by reconfiguration and linking of fragments with the same direction. In [18], Zhang *et al.* extracted global filament networks by using a line filter transform and an orientation filter transform to enhance filamentous features in single-molecule localization microscopy data. They then disconnected all filaments at

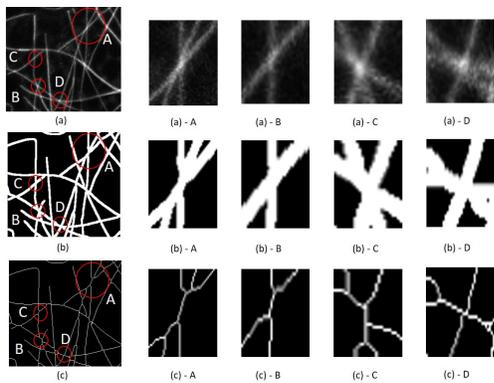


Figure 1. An example of area with complicated intersections. (a) Original image; (b) Semantic segmentation; (c) Skeletonization.

junction points and regrouped these fragments by the propagation vectors and distance vectors at each terminus of filament fragments. In [16], Zeder *et al.* proposed a new concept called fixel, which is a rectangle that covers a small part of a filament with similar orientation and is used to describe the smallest element of a filament. The image is first processed by a kernel to detect fixels, namely filamentous structures, and then filaments will be assembled by allocating fixels based on their orientation and location. All of these methods chose to dissect the filaments into fragments, and then regroup them according to their geometric properties. A common drawback of these methods is that it is hard for them to handle areas with complicated intersections, as shown in figure 1. These complicated junction areas usually cause false labeling of fragments. When the receptive field is small, it is hard to tell which fragments belongs to which filaments, since the propagation direction of pixels that close to the intersection can be very similar even if they belongs to different filaments. When the receptive field is too large, more filaments and more intersections can be involved and complicate the situation.

Recently, many approaches have applied deep learning methods to segment filamentous structures and have shown better semantic segmentation results than traditional methods. Saponaro *et al.* [13] used a modified U-net architecture [12] to segment vesicular networks of fungal hyphae in microscopy images and Liu *et al.* [8] used a stacked U-net architecture to segment microtubules in microscopy images. Fu *et al.* [2] combined a fully convolutional neural network with a fully-connected Conditional Random Field and achieved a better retinal vessel segmentation in fundus images. There are also some applications on non-biological thin, elongated structures. Zhang *et al.* [17] applied a modified U-Net architecture to segment road networks from aerial images. Fan *et al.* [1] proposed a convolutional neural network for segmenting pavement cracks. However, few methods can extract individual filaments from these seg-

mentation results, and it is not sufficient for domain experts to perform quantitative analysis on the biological samples. Thus, many popular filamentous structures analysis tools are mainly based on traditional methods [15, 18] which require manual tuning of parameters. This places a burden on the practitioner when processing huge volumes of images.

Many deep learning methods have been applied to instance segmentation, most of these methods can be considered as a conjunction between semantic segmentation and object detection. For example, in [5], He *et al.* proposed Mask R-CNN which extends Faster R-CNN by adding a branch to obtain the object mask which allowed for simultaneously detecting the objects in an image and segmenting each each instance. In [4], Hariharan *et al.* used multiscale combinatorial grouping to generate region proposals and obtain candidates, then applied R-CNN [3] to extract features of each region and performed non-maximum suppression on the candidates to remove duplicates. In the end, they use features from the CNN to produce coarse mask to refine the candidates, and combine the mask with original candidates to provide a more robust results. In [10, 11], their approaches integrated CNN with recurrent neural network, and by applying soft attention mechanisms, their models can sequentially find objects and segment them one at a time. Essentially, these instance segmentation models all require proposed candidate regions for individual objects. As mentioned in above, filaments are usually close in proximity, look similar, and have many overlapping and intersection areas between different objects, which makes it very difficult to extract region proposals for individual filaments.

The segmentation results in [13, 2, 17, 8] inspire us to find a way to utilize semantic segmentation results to obtain information about individual filaments. It is natural to adopt the strategies used in [14, 18], which is to extract the skeletons of semantic segmentation results, break them into fragments at intersection points, and then regroup the fragments into separate filaments. The most challenging part of this process is making sure that the fragments are correctly identified and regrouped at the intersections. We believe that the issue lies in trying to create fragments based on intersection points. To avoid fragmenting at the intersection points, we propose a deep neural network with multiple orientation-associated outputs for separating filaments based on their orientations. Each output is associated with fragments of filaments within a certain range of orientation angles, and thus separating them at intersections without disconnecting them. In other words, this network aims to turn an intersection into an overpass. However, since many filaments change directions as they propagate, filaments could be broken into fragments. The final step of our proposed approach is to connect these fragments according to their locations and propagation vector at their terminus, which is much less complex than handling connections at intersections.

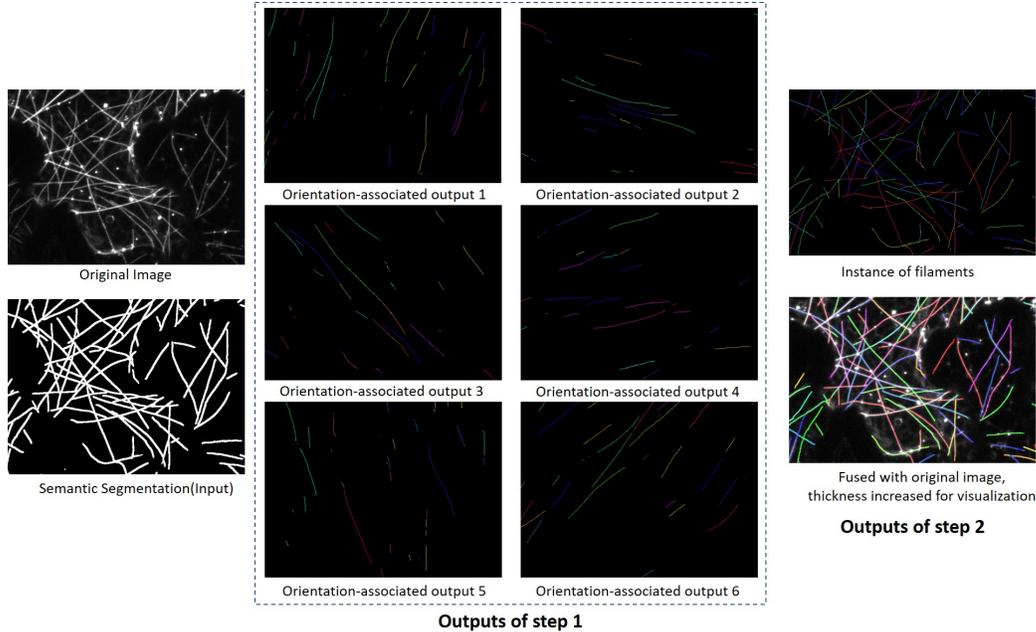


Figure 2. Pipeline of our proposed approach. Step 1: The network takes a segmentation as input and six orientation-associated outputs will be generated. Step 2: The proposed algorithm is applied on these six outputs and regroups fragments into full individual filaments.

In summary, the main contributions of this work are: (1) A Convolutional Neural Network that reformulates the problem of filament instance segmentation by outputting orientation-associated filament segments; (2) A terminus pairing algorithm for regrouping these fragments in different output layers, which achieves a remarkable performance on different datasets; (3) A dataset including 10 microscopy images of microtubule filament networks with each filament instance labeled.

## 2. Method

In this section, we describe our approach for segmenting individual filaments. Our approach includes two parts. First, we introduce the architecture of our network which decomposes the segmentation into six component layers. Each layer contains fragments with similar orientations. Then, we describe how to regroup fragments of the same filament from different layers. A pipeline of our proposed method is shown in Figure 2

### 2.1. Step One: Turning an Intersection into an Overpass

#### 2.1.1 Network Architecture

The U-net architecture [12] has achieved great success in segmenting vessel-like objects. We propose a new CNN architecture for extracting individual filaments at intersections, which is adapted from the U-net architecture (see

Figure 3). To address the problem of splitting filaments at intersections, we make two important modifications in our network. (a) We make six duplicate hourglass modules running in parallel, and each module is connected to orientation-associated ground truth maps. Each module focuses on an orientation range from  $[0^\circ, 30^\circ)$ ,  $[30^\circ, 60^\circ)$ ,  $[60^\circ, 90^\circ)$ ,  $[90^\circ, 120^\circ)$ ,  $[120^\circ, 150^\circ)$ ,  $[150^\circ, 180^\circ)$  respectively. We found empirically that six orientations are a good number to give enough angle resolution without over fragmenting. At intersections, fragments with different propagation directions will be sorted into different layers, which is like turning an intersection into an overpass. (b) The orientation-associated outputs of each module are concatenated together and fed into two paths. In the first path, the six output layers are merged together to form a prediction which is the skeletonized reconstruction of the input. This enables the six output layers to take information from the predictions of other modules into consideration, and enforce that the six output layers merged together result in the original filament network. In the second path, each output layer will be compared with predictions from the other modules with the constraint that the overlapping area between each be minimal. In summary, our network has 6 hourglass modules, each module works on splitting filaments within a specific range of orientations, the outputs of six modules together should reconstruct the original filamentous structure, and each output should be different from the outputs of other modules.

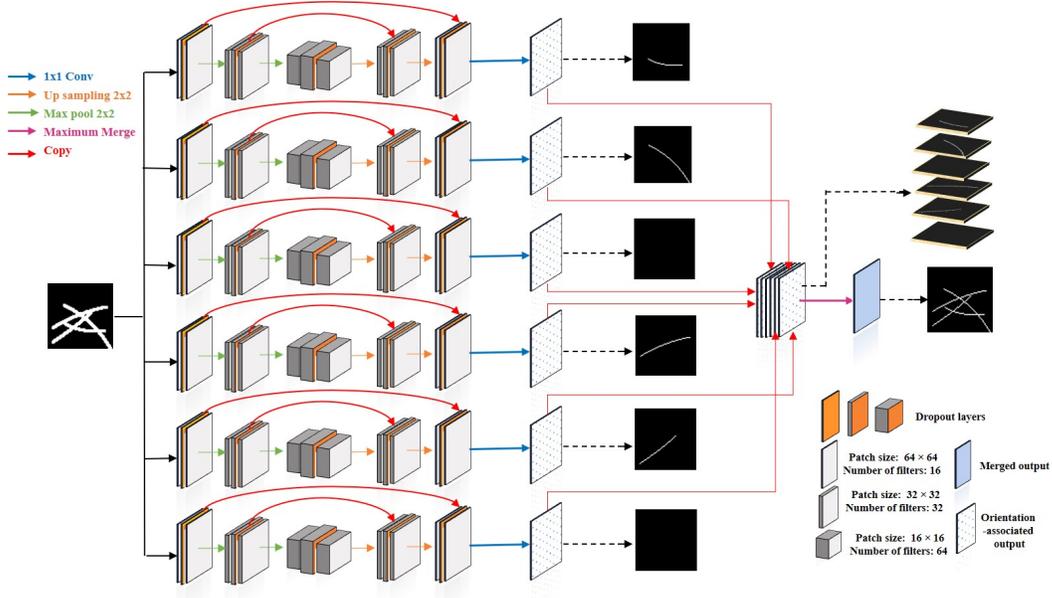


Figure 3. An illustration of our proposed network.

Each of the hourglass module mentioned above, contains a contracting path to capture features and an expansion path to retrieve localization information. There are five stages in total. Each stage contains two  $3 \times 3$  convolutional layers followed by a rectified linear unit and one drop out layer with a rate of 0.2 inserted between the two convolutional layers. In the contracting paths, a  $2 \times 2$  max pooling operation will be applied after each stage, and the number of feature channels will be doubled in the next stage. In expansion paths, each stage will be followed by a  $2 \times 2$  up-sampling operation, and halve the number of feature channels. Also, feature-map with the same size from contracting and expanding paths will be concatenated. At the end of each module, a  $1 \times 1$  convolutional layer is applied to obtain orientation-associated outputs. We use the dice coefficient loss for all orientation-associated outputs and the merged reconstruction output. Along with this, after the six outputs are concatenated together, a revised dice coefficient loss is used (overlap-loss). For each module we calculate the dice coefficient loss of the output with respect to the outputs of the remaining modules and take the average. Then sum up these six values and take the additive inverse of it, as we want the overlap between modules to be as small as possible. From our experiments, setting the weight of overlap-loss to 0.01 gave decent results. If the loss weight of overlap-loss is set too high, the network will not be able to separate filaments, but without this overlap-loss, unnecessary overlapping area can be created thus increasing the running time in Section 2.2.

### 2.1.2 Synthetic Dataset

As there is no sufficient dataset for our task, we create a synthetic dataset for training the network. First, we create 180000 straight lines with a range of random lengths. These lines are then deformed piece-wise by a random number of control points and random deformation scales. They are then offset by random translations. Each such generated strand is written as an image of size of  $64 \times 64$ . Then we evenly distribute these 180000 images into 6 buckets, 30000 in each bucket. For each bucket, we rotate the lines with a specific range of angles. From bucket 1 to bucket 6, the rotation angles are in range  $[0^\circ, 30^\circ)$ ,  $[30^\circ, 60^\circ)$ ,  $[60^\circ, 90^\circ)$ ,  $[90^\circ, 120^\circ)$ ,  $[120^\circ, 150^\circ)$ ,  $[150^\circ, 180^\circ)$  respectively. To create a training sample, we randomly select an image from each bucket. The selected images form the ground truth for each orientation-associated module in the network. We also have a randomized flag to decide whether or not to include a selected image. Ground truth for the orientation-associated module corresponding to the excluded image is set as a "blank" image. The selected and included images are then merged to form the ground truth for the combined reconstruction output of the network. To create the associated input, we dilate the merged layer with kernel size of 3, 5 and use it as input to the network. In total we create 120000 synthetic ground truth images with size of  $64 \times 64 \times 7$  (See Figure 4). Also, the ground truth images are not dilated because we also want our network to extract the skeleton of fragments.

The network architecture is fully convolutional, enabling

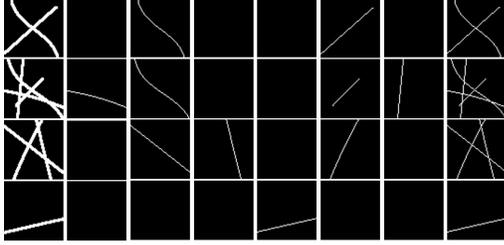


Figure 4. Examples of the synthetic dataset. From left to right: input image, orientation layer with range  $[0^\circ, 30^\circ)$ ,  $[30^\circ, 60^\circ)$ ,  $[60^\circ, 90^\circ)$ ,  $[90^\circ, 120^\circ)$ ,  $[120^\circ, 150^\circ)$ ,  $[150^\circ, 180^\circ)$ , and the final merged layer

us to train with small image patches, and thus we can use these synthetic images for training. There are several reasons for making synthetic images with small size: (a) It is hard to create synthetic images with a large image size because there will be too many filaments and the network of filaments can be very complicated. With small images, less filaments are involved, usually less than 6, and it is easier to create more realistic synthetic data. (b) With a larger image, the filaments tend to be longer and have more variation in orientation and curvature, which makes it harder to sort the filaments into bins of specific orientation ranges. Since our goal is to separate filaments at junctions, a smaller image size allows us to focus on the local area of an intersection to learn features useful for separating intersections. There is one downside of this approach: when the propagation direction of one filament changes significantly, this filament will be dissected into fragments and sorted to different layers (See Figure 5). In Section 2.2, we will discuss how to assemble these fragments back to individual filaments.

### 2.1.3 Training and Testing

We train our model for 5 epochs with a batch size of 64. The size of input patches is  $64 \times 64$ , and the number of training patches is 120000. For each module, the numbers of feature channels are 16, 32, 64 for corresponding stages. All networks are trained using Adam optimizer with a learning rate of 0.0001. Dropout rates of all dropout layers are set to 0.2. For our proposed network, we compile the model and assign a weight of 1 for all dice coefficient loss, and 0.01 for overlap-loss. The model achieved a validation IoU of 0.793 averaged across the 6 outputs and a validation IoU of 0.97 on the combined reconstruction. All experiments were conducted on two NVIDIA GeForce Titan X (Pascal) GPUs.

## 2.2. Step Two: Regrouping Fragments

As shown in Figure 5, some filaments could be dissected into several fragments and distributed into different output

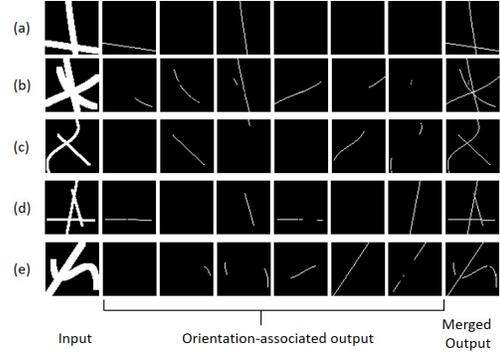


Figure 5. Output examples of our proposed network. Input images are synthetic test images. (a): Two filaments perfectly separated. (b),(c),(d),(e): Some large filaments are dissected into different layers and some fragments appear in more than one layer, e.g the horizontal filaments in (d).

layers. In this step, filaments are reconstructed by connecting fragments according to the orientation and spatial location of their termini. First, a local search in six layers is carried at the termini of each filament, and all fragments within the search area will be listed as candidates. The next step is to find out the best matching candidate and group it to the selected fragment and update the selected fragment. This process will be done recursively until there is no matching candidate, as shown in Figure 6. To find out the best matching candidate, there are two major criteria: overlapping case and non-overlapping case. Overlapping cases (see Figure 7) will be considered first and the matching rules of overlapping cases are as follows: (1) If a candidate that is fully covered by the selected fragment or the selected fragment is fully covered by the candidate, the candidate will be grouped to selected filament. (2) If the candidate is partially overlapping with the selected fragment, the eligibility of the fragment candidates is determined by the spatial location of termini and the end points of the overlapping area. As shown in Figure 7, if one terminus of one fragment lies on the other fragment, and vice versa, and these two termini are also the end points of overlapping area, then these two fragments should be grouped together. If there are no eligible overlapping cases, the following criteria (see Figure 8) will be considered: (1) Similarity between propagation direction of termini of selected filament and candidates. (2) Distance between termini of selected filament and candidates. (3) Similarity between propagation direction of termini and the vector from the termini of selected filament to the termini of candidate. Thresholds will be set for each criteria, and we only consider candidates that satisfy all three thresholds. The most eligible matching pair will be the pair of fragments with the most similar propagation vectors, then the most similar distance vectors and then the shortest distance.

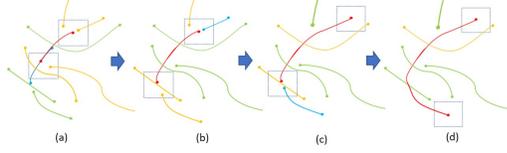


Figure 6. An example of recursively grouping the most eligible candidate. The blue box is the searching area; solid circles are termini of fragments; red represents the selected fragment; blue represents the best eligible candidate; yellow represents candidates or fragments detected in the search box, which does not indicate they are eligible fragments; green represents irrelevant fragments. (a) group overlapping fragments, overlapping fragments with highest priority; (b), (c) group the most eligible fragment; (d) no more eligible fragments.

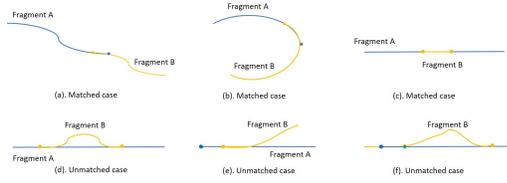


Figure 7. Examples of handling overlapping cases. (a), (b) Matched. One terminus of A is the end point of an overlapping area and lies on B and vice versa; (c) Matched. A fully covers B; (d) Unmatched. B's two termini lie on A, but B is not fully covered by A; (e) Unmatched A's termini does not lie on B. (f) Unmatched termini of A B are not endpoints of overlapping (e.g the green point).

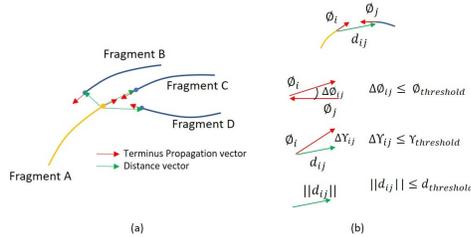


Figure 8. (a) The most eligible candidate will be grouped. In this example, C will be grouped with A. (b) A matching criteria. After applying thresholds, invalid candidates will be filtered out.

### 3. Experimental results

In this section, we will discuss about the experiments we have run, and compare the performance of our proposed methods with other approaches.

#### 3.1. Evaluation protocol

To evaluate the performance of individual filaments extraction, we first adopt the metric of Skeletonized Intersection over Union (SKIoU) introduced in [8], which is defined

as:

$$SKIoU = \frac{2 * Skel (Intersection)}{Skel (Prediction) + Skel (Ground Truth)}$$

where *skel* is Skeletonization operation

(1)

Similar to IoU metric, the numerator is the number of pixels of the skeletonization of the overlapping area times two. The number of pixels of the skeletonized prediction plus the number of pixels of the skeletonized ground truth will be denominator. SKIoU metric is very suitable for filamentous structures as it is less sensitive than IoU and it focuses on the length and spatial location of the objects. Every filament in the ground truth image will be compared with each filament in the output image. The filaments with maximum SKIoU scores will be considered as matching pairs and each filament of both predicted filaments and ground truth filaments can only belong to at most one pair. After we obtain the SKIoU score for each filaments, we evaluate the average precision, recall and F-measure (Equation 2) score using SKIoU thresholds ranging from 0.5 to 0.95 with 0.05 increments.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

(2)

#### 3.2. Microtubule dataset

Microtubules are very important filamentous structures in a number of cellular processes. To quantify our result we collect a dataset of 10 microscopy images of microtubules with image size  $122.03 \times 132.84$  ( $1400 \times 1524$  pixels). The number of microtubules in each image is more than 300, so it is extremely hard to label individual filaments. To reduce the workload of manual labeling, we asked domain experts to manually correct the output of our approach. Each image takes around 2 hours to correct. More images will be manually labeled in the future, and we will share our dataset online.

Since our network takes segmentation result as input, we adopt the Stacked U-net architecture in [8] and obtain the segmentation. These segmentation results are used as input for our approach.

We also have evaluated methods in [15, 18] on our dataset for comparison. Since the method in [18] is designed for a different type of image, it is not fair to directly apply their methods on the images we collected. So we only apply their regrouping methodology on our segmentation output to extract individual filaments. Parameters of these methods [15, 18] are adjusted as recommended. It should be mentioned that SOAX is designed for actin filaments, so it is reasonable that SOAX doesn't work well on microtubules. Since SOAX creates many small fragments

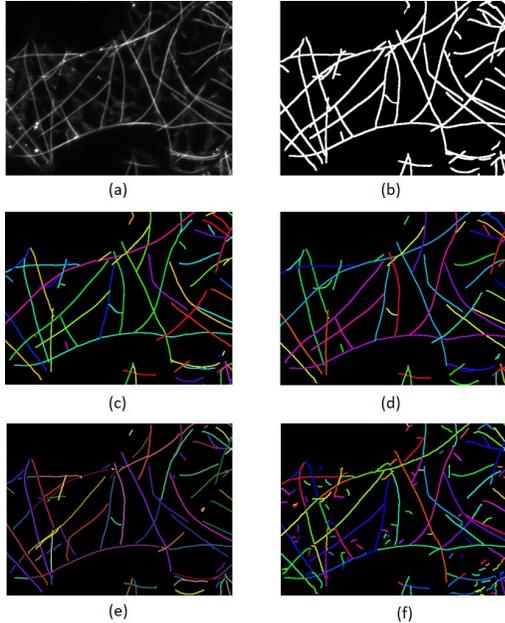


Figure 9. Comparison of different methods on extracting individual filaments. (a) Original image (b) Segmentation (c) Manually corrected labels (d) Result of our proposed method (e) SFINE (f) SOAX without filtering out small fragments. Different colors indicate different labels.

Method	F1	P	R	Ave SKIoU
SOAX[15]	0.2691	0.2087	0.4196	0.447
SIFNE[18]	0.2875	0.2458	0.3623	0.5604
Ours	0.7161	0.6456	0.8189	0.8166

Table 1. Comparison between different methods for Microtubule extraction on our microtubule dataset. F1 is F-measure; P is precision; R is recall, and they are evaluated using SKIoU thresholds ranging from 0.5 to 0.95 with 0.05 increments. Ave SKIoU is the average skIoU over all detected filaments.

(see Figure 9), we filtered out those fragments shorter than 25 pixels, and then conducted the quantification analysis.

Quantitative comparison are shown in table 1, and Figure 9 shows a qualitative comparison between different approaches. Our approach has shown better performance over all metrics. Since our dataset is manually corrected from our outputs, the results in table 1 can be biased to our method. But as shown in Figure 9, our method has successfully extracted many long filaments, and there are less false regrouping cases. SOAX [15], and SIFNE[18] both fail to reconnect many fragments at intersections, especially when several filaments intersect with each other within a small region. Because SOAX and SIFNE dissect filaments into more fragments and fail to reconnect them, the SKIoU and F-measure scores of these methods tend to be much lower.

Figure 10 has shown an example of the complexity of

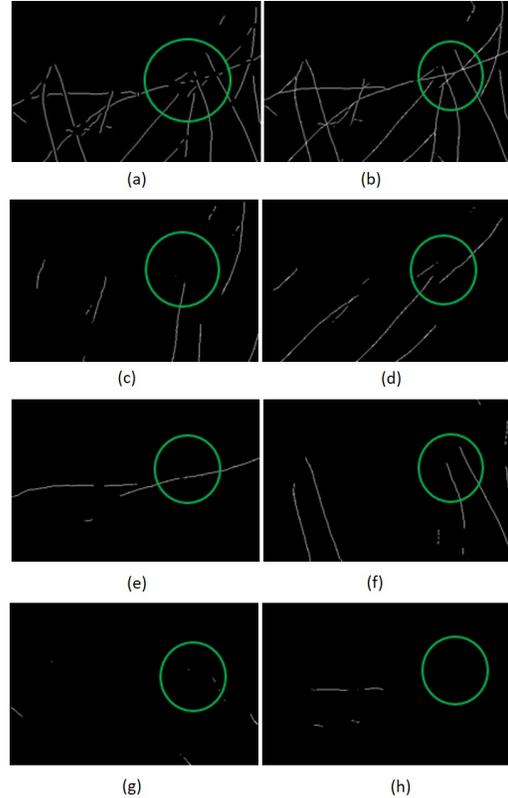


Figure 10. (a): An intermediate step of SFINE, which is after extracting out all junction areas and before regrouping. (b): Merged output from our proposed network. (c)-(h): orientation-associated outputs. Our pairing algorithm deals with simpler situations.

regrouping fragments at intersections and the difference between SFINE and our method. Regards running times, SOAX takes around six hours to run on a whole image obtaining segmentation and individual filaments; SFINE takes around 4 minutes to extract individual filaments from segmentation results, while ours takes approximately 30 minutes for each image. Our methods can handle complicated intersections better, which leads to a higher accuracy of extracting individual filaments.

### 3.3. Microtubule dataset provided by Zeder *et al.*

We have tested our method on the microtubule dataset provided in [16], which includes 7 images with size of  $5000 \times 5000$  pixels. The dataset includes around 192 filaments, and we applied the same metric used in [16]. Table 2 shows the quantitative results and Figure 11 shows the qualitative results. Our method successfully extracts filaments in most cases (Figure 11), and Zeder’s approach has also achieved high accuracy on the dataset. As shown in table 2, our result is slightly better than Zeder’s method regard to number of filaments, but it doesn’t mean that our method outperforms

Image	Manual Count[16]	Zeder's Method[16]	Ours
1	25.8 ± 1.5	27	25
2	18.5 ± 0.5	23	22
3	27.5 ± 1.5	27	26
4	44.0 ± 0	44	44
5	36.8 ± 0.4	40	37
6	21.8 ± 0.4	21	24
7	17.5 ± 1.1	18	18
Total	191.8 ± 3.6	200	<b>196</b>

Table 2. Quantified results on the dataset in work [16]. Manual and automated counts given in [16] and our results.

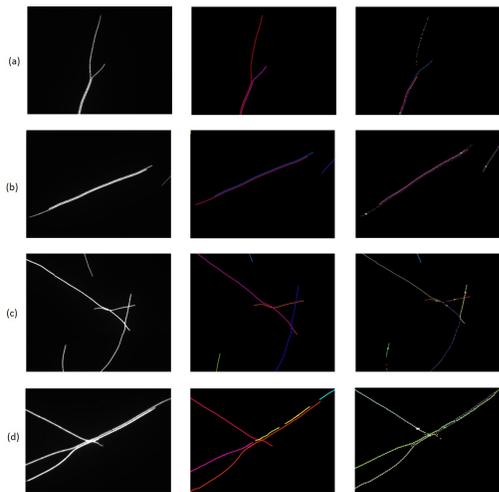


Figure 11. Qualitative results on the dataset in work [16]. From left to right: The original image, our method, and Zeder's approach. (a) Touching case. (b) Parallel case. (c) Intersection and overlapping case (d) Failure case of our approach

Zeder's method. As shown in Figure 11 (d), our approach fails to reconnect the yellow, pink and blue fragment while Zeder's approach grouped 3 filaments successfully in the image. Two reasons can cause this failure: 1. Segmentation result is not correct. 2. The parameters like searching range are not appropriately set in the regroup step. In Figure 11 (c), our approach successfully segment out three filaments that intersect with each other, but Zeder's methods fails to group them correctly at the intersections.

### 3.4. Road Network Dataset

Because our approach takes semantic segmentation as input, it can be applied to other similar structures, like road networks and fibers. As long as there are decent semantic segmentation. In this section, we applied our network on Road Detection Datasets [9]. Though roads are very different from microtubules, they are both elongated thin objects, and in most cases, roads doesn't branch. We adopt

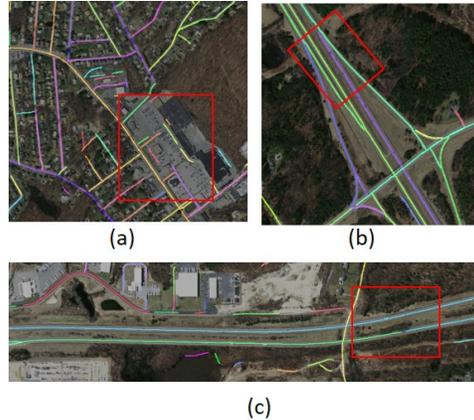


Figure 12. Examples of extracting individual roads on Road Detection Datasets. Red boxes highlight failed extractions.

the stacked U-net architecture in [8] to obtain segmentation first, which has achieved an average IoU of 0.64 and skIoU 0.87 on this dataset. Since this dataset does not provide individual road labeling, we are only going to show qualitative results of our experiment.

### 3.5. Limitations

There are several limitations of our approach. 1. Under some cases, the six side outputs merge together may not fully recover the original input, which causes large gaps between fragments and leads to failure in regroup step (Example: Figure 11 (d) ). 2. The proposed terminus pairing algorithm may fail to find the right pairs. This is because the best match based on the criteria might not be the right one (Example: Red box in Figure 12 (b)). 3. Since our work processes on segmentation results, our approach will not work if segmentation is not accurate.

## 4. Conclusion

We proposed a fully convolutional neural network with multiple orientation-associated side outputs to address the problem of extracting individual filaments. Each branch is able to detect fragments with a specified range of propagation directions, and thus extracting different filaments at the intersection region. Additionally, we propose an algorithm to regroup these fragments and reconstruct individual filaments. Our proposed network enable us to avoid regrouping fragments at intersections, and thus improves the performance. We have tested our method on a microtubule dataset we created, a public microtubule dataset, and a road dataset. Proposed approach outperforms other approaches on our dataset, and it has shown outstanding results on other datasets.

## References

- [1] Z. Fan, Y. Wu, J. Lu, and W. Li. Automatic pavement crack detection based on structured prediction with the convolutional neural network. *arXiv preprint arXiv:1802.02208*, 2018.
- [2] H. Fu, Y. Xu, D. W. K. Wong, and J. Liu. Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 698–701. IEEE, 2016.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [4] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.
- [5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.
- [6] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [8] Y. Liu, W. Treible, A. Kolagunda, A. Nedo, P. Saponaro, J. Caplan, and C. Kambhamettu. Densely connected stacked u-network for filament segmentation in microscopy images. In *European Conference on Computer Vision*, pages 403–411. Springer, 2018.
- [9] V. Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto, 2013.
- [10] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pages 21–26, 2017.
- [11] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European Conference on Computer Vision*, pages 312–329. Springer, 2016.
- [12] O. Ronneberger, P. Fischer, and T. 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.
- [13] P. Saponaro, W. Treible, A. Kolagunda, T. Chaya, J. Caplan, C. Kambhamettu, and R. Wisser. Deepxscope: Segmenting microscopy images with a deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 91–98, 2017.
- [14] T. Xu, D. Vavylonis, and X. Huang. 3d actin network centerline extraction with multiple active contours. *Medical image analysis*, 18(2):272–284, 2014.
- [15] T. Xu, D. Vavylonis, F.-C. Tsai, G. H. Koenderink, W. Nie, E. Yusuf, I.-J. Lee, J.-Q. Wu, and X. Huang. Soax: a software for quantification of 3d biopolymer networks. *Scientific reports*, 5:9081, 2015.
- [16] M. Zeder, S. Van den Wyngaert, O. Köster, K. M. Felder, and J. Pernthaler. Automated quantification and sizing of unbranched filamentous cyanobacteria by model-based object-oriented image analysis. *Applied and environmental microbiology*, 76(5):1615–1622, 2010.
- [17] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 2018.
- [18] Z. Zhang, Y. Nishimura, and P. Kanchanawong. Extracting microtubule networks from superresolution single-molecule localization microscopy data. *Molecular biology of the cell*, 28(2):333–345, 2017.