

UC Santa Barbara

UC Santa Barbara Previously Published Works

Title

Accurate 3D Cell Segmentation Using Deep Features and CRF Refinement

Permalink

<https://escholarship.org/uc/item/9dk368f2>

Authors

Jiang, Jiaxiang
Kao, Po-Yu
Belteton, Samuel A
[et al.](#)

Publication Date

2019-01-25

DOI

10.1109/icip.2019.8803095

Peer reviewed


ACCURATE 3D CELL SEGMENTATION USING DEEP FEATURE AND CRF REFINEMENT

Jiaxiang Jiang* Po-Yu Kao* Samuel A. Belteton[†] Daniel B. Szymanski[†] B.S. Manjunath*

* Department of Electrical and Computer Engineering, University of California, Santa Barbara

[†] Department of Botany and Plant Pathology, Purdue University

ABSTRACT

We consider the problem of accurately identifying cell boundaries and labeling individual cells in confocal microscopy images, specifically, 3D image stacks of cells with tagged cell membranes. Precise identification of cell boundaries, their shapes, and quantifying inter-cellular space leads to a better understanding of cell morphogenesis. Towards this, we outline a cell segmentation method that uses a deep neural network architecture to extract a confidence map of cell boundaries, followed by a 3D watershed algorithm and a final refinement using a conditional random field. In addition to improving the accuracy of segmentation compared to other state-of-the-art methods, the proposed approach also generalizes well to different datasets without the need to retrain the network for each dataset. Detailed experimental results are provided, and the source code is available on GitHub ¹.

Index Terms— Cell segmentation, convolutional neural networks, 3D U-Net, 3D watershed, conditional random field

1. INTRODUCTION

Accurate cell boundary detection from 3D confocal imagery is critical in many applications including modeling of cell morphogenesis [1] and plant growth [2]. A typical cell segmentation workflow consists of the following steps: (i) image pre-processing to enhance the signal-to-noise ratio, (ii) a segmentation method that then partitions the cells, and (iii) a final refinement step. Typical pre-processing methods including morphological operations, edge enhancement methods, and denoising are widely used in the recent cell segmentation tasks [3, 4, 5, 6]. This pre-processing stage is often data-dependent and requires parameter tuning to avoid under- or over-segmentation. For membrane-tagged images, segmentation is usually carried out with either 3D watershed based methods [2, 4] or 3D level sets [7]. Some methods work on individual 2D images and then fuse the results to get a 3D segmentation [3, 8]. Finally, post-processing using cell volume or shape heuristic is usually applied. For this traditional

workflow of cell segmentation, pre-processing is subjective, and its parameters are highly dependent on the data.

During the past few years, there has been much interest in the use of deep learning for the cell segmentation problems, and the U-Net in particular has been widely explored [9, 10]. U-Net is a fully convolutional network which consists of encoder and decoder parts. The encoder part uses context information to encode raw images into feature maps, and the decoder part uses the produced feature maps to localize objects and generate the segmentation. U-Net based method gives fairly good cell boundary segmentation accuracy on the similar dataset but its performance drops in generalizing to other datasets. Besides, U-Net is not guaranteed to generate closed cell surfaces.

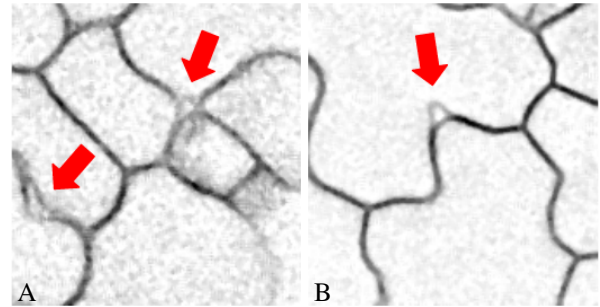


Fig. 1. (A) Inter-cellular spaces and (B) *Protrusion* are indicated by red arrows. **Best viewed in color.**

In this paper, we aim to solve the problem of accurately identifying cell boundaries and labeling individual cells in confocal microscopy image stacks. The goal is to generate closed cell surfaces in the 3D image stack while being able to accurately delineate features of interest such as the inter-cellular spaces and *protrusions*, see fig. 1. The first step is to generate a membrane probability map where voxels that are likely to belong to a cell membrane have higher values than those that are not on the cell membrane. This is achieved by a 3D U-Net architecture which is trained on a large, public data. This is followed by a watershed method that labels individual cells. One challenge for watershed algorithm is automated seeding of the cells, and this is achieved using a simple distance transform of the membrane probability map. The watershed based approach is sensitive to the image signal and likely

This work was supported by NSF MCB Grant No. 1121893 to D.B.S. and NSF MCB Grant No. 1715544 to B.S.M.

¹<https://github.com/UCSB-VRL/Purdue3DCell>

to smooth out the boundaries and also miss smaller features that are significant in modeling cell morphogenesis. For this purpose, we introduce a final processing step based on conditional random fields (CRFs). The CRF refines the watershed boundaries taking into account the probability map and local voxel boundary information. As demonstrated through the extensive experiments, this would help the overall approach to be sensitive to local boundary perturbation, including detecting salient boundary features. The proposed method is validated in two datasets, and the experimental results shows that our segmentation method outperforms the current cell segmentation methods [2, 11, 12] with less computation time. In addition, we also demonstrate that the proposed method generalizes well to different datasets without addition retraining or fine tuning the parameters of the neural network.

To summarize, the main contributions of this paper include (i) a method that is sensitive to local image features, for accurate detection and localization of boundaries in 3D confocal stacks; (ii) and show that the method is capable of generalizing to other datasets without further training.

2. MATERIAL AND METHODS

2.1. Dataset

Two 3D confocal image stack datasets of fluorescent-tagged plasma-membrane cells are used in this paper. In both datasets, only the plasma-membrane signal is available and is represented by voxels with high intensity values. The first dataset contains 124 image stacks of cells in the shoot apical meristem of 6 *Arabidopsis thaliana* [13]. In each image stack, there are 150 to 300 slices containing of 3 layers of cells: outer layer (L1), middle layer (L2), and deep layer (L3), and the size of each slice is 512×512 . The available resolution of each image in x and y direction are $0.22\mu m$ and in z is about $0.27\mu m$. We should note that as we go from L1 towards L3, the image quality degrades progressively due to scattering of light. This condition makes the segmentation problem challenging in L3. In this dataset, the ground truth voxel-wise cell labels are provided, and each cell has a unique label. The second dataset [1] consists of a long-term time-lapse from *A. thaliana*'s leaf epidermal tissue that spans over a 12 hour period with a xy-resolution of $0.212\mu m$ and $0.5\mu m$ thick optical sections. There are 20 image stacks in this dataset. In each image stack, there are 15 to 30 slices containing one layer of cell, and the size of each slice is 512×512 . The ground truth cell labels are not provided in this dataset.

2.2. Method

Our method for the 3D cell segmentation contains three steps as shown in fig. 2. First, a 3D U-Net based neural network is used to generate a probability map of each voxel being the membrane. Second, a 3D watershed algorithm whose seeds

are generated automatically is applied to this probability map, and outputs the initial cell segmentation result. Finally, a CRF is used to refine the boundaries of this initial cell segmentation.



Fig. 2. The general workflow of our proposed method.

2.2.1. Pre-processing

First, we apply a background subtraction on the raw cell image using the rolling ball algorithm [14]. This algorithm estimates the intensity value of background by averaging over a large neighborhood (ball) of each voxel. Then, we subtract this background from the raw cell image. Next, spatial resolution adjustment and histogram matching are applied to this background-subtracted cell image that accommodate different spatial resolution and different voxel intensity distribution of different datasets.

2.2.2. Membrane Probability Map using U-Net

For probability map generation, a 3D U-Net based network is used. For training the network, we used the *A. thaliana* dataset [13]. Half the dataset is used for training and the other half for testing, and we only use the L1 layer cells from training set to train the network. The architecture of our model is based on the modified 3D U-Net [15] with group normalization [16]. In our model, the input are the sub-slices of 3D stack images (of dimensions $512 \times 512 \times 16$). Due to computational limitations, we are able to input only 2 batches at a time. With this small batch size, standard batch normalization is unstable and tends to have a higher error so we use group normalization instead [17]. In our implementation, the default number of groups is set to 4. We use the Huber loss function for training, defined as:

$$L_{\delta}(y, \hat{y}(I)) = \begin{cases} \frac{1}{2}(y - \hat{y}(I))^2, & \text{if } |y - \hat{y}(I)| < \delta \\ \delta|y - \hat{y}(I)| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (1)$$

where $y \in \{0, 1\}$ is ground truth label representing whether the voxel is membrane, I is the input image, and $\hat{y} \in [0, 1]$ is the regression function which is learned by the 3D U-Net. The reason we use Huber loss is that the network is used to learn the regression function but not the classification function. This loss function is quadratic when the predicted output is close to the ground truth and linear when the prediction is far from the ground truth. δ determines the threshold value between quadratic and linear loss. This loss function is differentiable compared to a mean absolute loss function and less sensitive to outliers than a mean squared loss function.

Fig. 3(B) shows an example of the membrane probability map generated by the 3D U-Net.

2.2.3. 3D Watershed Segmentation

A standard 3D watershed segmentation algorithm is now applied to the 3D probability map. The challenging part of the watershed segmentation is to find seeds of each cell. In most cell images with a stained or fluorescent-tagged nucleus, the nuclei can be used as seeds. However, in surface labeled cell images, nuclei information is not tagged and the seed points need to be automatically generated.

Towards this, we use a standard Otsu’s thresholding on the probability map to get a binary image, and then compute a 3D distance transform on this binary image. The distance transform gives the minimum distance with respect to membrane for each voxel. In order to generate one seed within each cell, a H-maxima transform [18] is applied. The H-maximum transform suppresses all maxima in this distance transform map whose value is less than a threshold H compared to surrounding voxel values. The remaining maxima are the seed locations, to which the 3D watershed method is applied. Fig. 3(C) shows the result of 3D watershed segmentation.

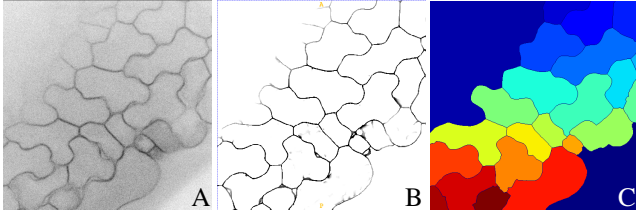


Fig. 3. (A) Inverted raw image in xy orientation, (B) inverted probability map from the 3D U-Net, (C) initial segmentation result from 3D watershed. **Best viewed in color.**

2.2.4. CRF Refinement

In the final step we use a dense CRF [19] to refine the segmentation boundaries. For the given probability map $\mathbf{Q} = \hat{y}(I)$ from eq. 1 and initial cell labels \mathbf{X} , the conditional random field is modeled by the Gibbs distribution,

$$P(\mathbf{X}|\mathbf{Q}) = \frac{1}{Z(\mathbf{Q})} \exp(-E(\mathbf{X}|\mathbf{Q}))$$

where denominator $Z(\mathbf{Q})$ is the normalization factor. The exponent is the Gibbs energy function (for notation convenience, all conditioning is omitted from this point for the rest of the paper) and we need to minimize the energy function $E(\mathbf{X})$ to get the final refined label assignments. In the dense CRF model, the energy function is defined as

$$E(\mathbf{X}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \quad (2)$$

where x_i and x_j are the vertices of CRF, and i and j are the indices of each voxel which iterate over all voxels in the graph. $i, j \in \{1, 2, \dots, N\}$ and N is the total number of voxels in the image stack. $x_i, x_j \in \{1, 2, \dots, L\}$ and L is the total number of cells (seeds) identified by the watershed method. The first term of eq. 2, the unary potential, is given by $\psi_u(x_i) = -\log P(x_i)$, where $P(x_i)$ is the probability of voxel i having the specific label x_i . It is calculated based on the probability map and the label image of the watershed. $P(x_i) = 1 - q_i$ if voxel i is inside the cell with label x_i after the watershed, and $P(x_i) = 0$ otherwise. q_i is the i th voxel value in the probability map. $1 - q_i$ represents the probability of voxel being the interior point of the cell. The pairwise potential in eq. 2 takes into account the label of neighborhood voxels to make sure the segmentation label is closed and the boundary is smooth. It is given by:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_m w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)$$

where the penalty term $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, and $\mu(x_i, x_j) = 0$ otherwise. $w^{(m)}$ is the weight for each segmentation label $m \in \{1, 2, \dots, L\}$, and $k^{(m)}$ is the pairwise kernel term for each pair of voxels i and j in the image stack regardless of their distance that capture the long-distance voxel dependence in the image stack. \mathbf{f}_i and \mathbf{f}_j are feature vectors from the probability map. \mathbf{f}_i incorporates location information of voxel i and the corresponding value in the probability map ($\mathbf{f}_i = \langle \mathbf{p}_i, q_i \rangle$ where $\mathbf{p}_i = \langle x_i, y_i, z_i \rangle$, and x_i, y_i and z_i are the voxel i in x, y , and z coordinate). Specifically, the kernel k is defined as

$$w_1 \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\alpha^2} - \frac{\|q_i - q_j\|^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\gamma^2}\right)$$

where the first term depends on voxel location and the corresponding voxel value in probability map. The second term only depends on the voxel location. $\sigma_\alpha, \sigma_\beta$, and σ_γ are the hyper parameters that depends on shape of cells in each dataset. Fig. 4 shows an example of the results before and after CRF refinement.



Fig. 4. (A) Inverted probability map from the 3D U-Net, (B) initial segmentation without CRF refinement, (C) final segmentation result with CRF refinement. **Best viewed in color.**

3. EXPERIMENTAL RESULTS

We apply our cell segmentation method on 3D confocal image stacks of *A. thaliana* shoot apical meristem cells [13]. The

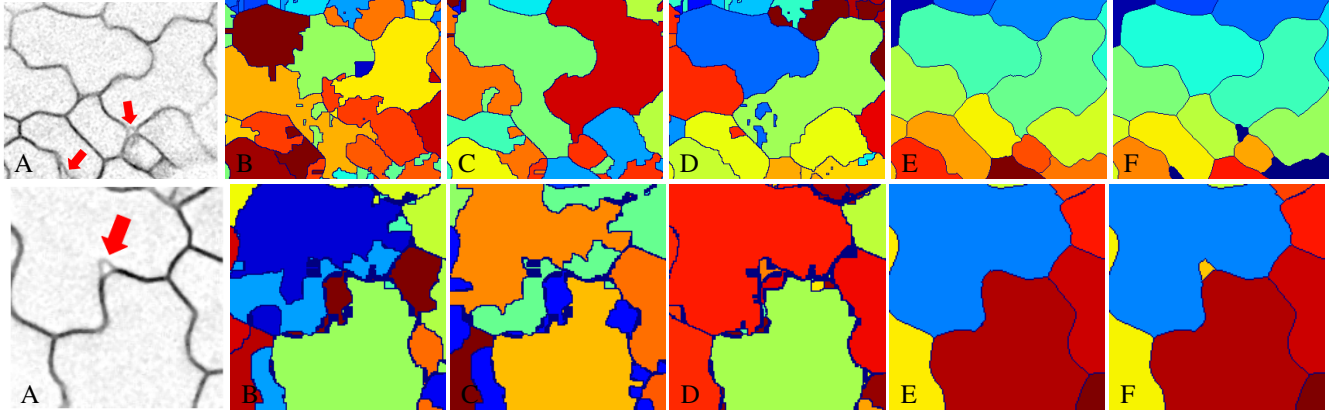


Fig. 5. The top row shows the segmentation results of the cell image with inter-cellular spaces indicated by red arrows and the bottom row shows the segmentation results of the cell image with a *protrusion* pointed by a red arrow. (A) Inverted raw image in xy orientation, (B) MARS, (C) ACME, (D) supervoxel-based method, (E) proposed method without CRF, (F) proposed method. **Best viewed in color.**

L_1 layer of the training set is used to train the 3D U-Net, and L_1 , L_2 and L_3 layers of the testing set are used to evaluate the segmentation performance of the proposed algorithm. Table 1 to 3 show the comparison of the final result using our proposed method and other methods including ACME [12], MARS [11] and a supervoxel-based algorithm [2] on L1 to L3 respectively. With CRF refinement, our model improves approximately 0.028 in the F-score measure on average. In addition, our proposed model has the best F-score measure on L1, L2 and L3 compared to other methods. It is noted that the average segmentation time of our proposed model is significantly shorter compared to the supervoxel-based method [2]. Our proposed method takes approximately 0.6 seconds to segment one slice of image stacks on average, whereas supervoxel-based method takes approximately 6 seconds on a NVIDIA GTX Titan X and an Intel Xeon CPU E5-2696 v4 @ 2.20GHz.

In the second experiment, we apply our proposed method to the second dataset [1] for the purpose of identifying and analyzing cells based on the segmentation. The segmentation results of our proposed method and other current methods are shown in fig. 5. Although the quantitative results are not available in this dataset, our proposed method visually has better segmentation performance, and our method is able to identify the inter-cellular spaces and *protrusions* in the 3D cell image stack. For these image stacks, the average number of cells in the image is 25.20 ± 10.20 , and the average detected cells using our proposed method is 25.10 ± 10.30 which is better than MARS (35.10 ± 15.29), ACME (20.06 ± 8.30) and the supervoxel method (27.78 ± 11.30). For the cell counting task, CRF refinement does not change the number of the cells since it only modifies the boundary of the cell. However, with CRF refinement, we are able to get more accurate cell boundaries.

Table 1. 3D Segmentation Performance on L1

| Algorithm | Precision | Recall | F-Score |
|-------------------|--------------|--------------|--------------|
| ACME | 0.805 | 0.966 | 0.878 |
| MARS | 0.910 | 0.889 | 0.899 |
| Supervoxel method | 0.962 | 0.932 | 0.947 |
| Result before CRF | 0.944 | 0.930 | 0.937 |
| our method | 0.953 | 0.973 | 0.963 |

Table 2. 3D Segmentation Performance on L2

| Algorithm | Precision | Recall | F-Score |
|-------------------|--------------|--------------|--------------|
| ACME | 0.775 | 0.980 | 0.866 |
| MARS | 0.921 | 0.879 | 0.900 |
| Supervoxel method | 0.910 | 0.932 | 0.921 |
| Result before CRF | 0.924 | 0.920 | 0.922 |
| our method | 0.943 | 0.973 | 0.953 |

Table 3. 3D Segmentation Performance on L3

| Algorithm | Precision | Recall | F-Score |
|-------------------|--------------|--------------|--------------|
| ACME | 0.745 | 0.976 | 0.845 |
| MARS | 0.909 | 0.879 | 0.894 |
| Supervoxel method | 0.982 | 0.881 | 0.929 |
| Result before CRF | 0.933 | 0.888 | 0.910 |
| our method | 0.943 | 0.932 | 0.937 |

4. CONCLUSION

We present a probability map based 3D cell segmentation method which requires very few parameters for membrane tagged images. This method contains a probability map generation, a 3D watershed transform, and a CRF refinement. Our experiments used pavement cell data [1] and *A. thaliana* [13] dataset, and quantitative results show that the proposed method achieves better segmentation performance compared to other current methods with much less computation time.

5. REFERENCES

- [1] Samuel A Belteton, Megan G Sawchuk, Bryon S Donohoe, Enrico Scarpella, and Daniel B Szymanski, “Re-assessing the roles of pin proteins and anticyclinal microtubules during pavement cell morphogenesis,” *Plant Physiology*, vol. 176, no. 1, pp. 432–449, 2018.
- [2] Johannes Stegmaier, Thiago V Spina, Alexandre X Falcão, Andreas Bartschat, Ralf Mikut, Elliot Meyerowitz, and Alexandre Cunha, “Cell segmentation in 3d confocal images using supervoxel merge-forests with cnn-based hypothesis selection,” in *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*. IEEE, 2018, pp. 382–386.
- [3] Johannes Stegmaier, Fernando Amat, William C Lemon, Katie McDole, Yinan Wan, George Teodoro, Ralf Mikut, and Philipp J Keller, “Real-time three-dimensional cell segmentation in large-scale microscopy data of developing embryos,” *Developmental cell*, vol. 36, no. 2, pp. 225–240, 2016.
- [4] Romain Fernandez, Pradeep Das, Vincent Mirabet, Eric Moscardi, Jan Traas, Jean-Luc Verdeil, Grégoire Malandain, and Christophe Godin, “Imaging plant growth in 4d: robust tissue reconstruction and lineaging at cell resolution,” *Nature methods*, vol. 7, no. 7, pp. 547, 2010.
- [5] Jun Kong, Fusheng Wang, George Teodoro, Yanhui Liang, Yangyang Zhu, Carol Tucker-Burden, and Daniel J Brat, “Automated cell segmentation with 3d fluorescence microscopy images,” in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*. IEEE, 2015, pp. 1212–1215.
- [6] Sorin Pop, Alexandre C Dufour, Jean-François Le Garrec, Chiara V Ragni, Clémire Cimper, Sigolène M Meilhac, and Jean-Christophe Olivo-Marin, “Extracting 3d cell parameters from dense tissue environments: application to the development of the mouse heart,” *Bioinformatics*, vol. 29, no. 6, pp. 772–779, 2013.
- [7] Yiyi Chen, Hanxu Sun, Huihua Yang, and Xipeng Pan, “Level set method of cell image segmentation based on combinations of edge, region and prior information,” in *Systems and Informatics (ICSAI), 2017 4th International Conference on*. IEEE, 2017, pp. 1245–1249.
- [8] Diana L Delibaltov, Utkarsh Gaur, Jennifer Kim, Matthew Kourakis, Erin Newman-Smith, William Smith, Samuel A Belteton, Daniel B Szymanski, and BS Manjunath, “Collect: cell evolution capturing tool,” *BMC bioinformatics*, vol. 17, no. 1, pp. 88, 2016.
- [9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [10] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf 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.
- [11] Romain Fernandez, Pradeep Das, Vincent Mirabet, Eric Moscardi, Jan Traas, Jean-Luc Verdeil, Grégoire Malandain, and Christophe Godin, “Imaging plant growth in 4d: robust tissue reconstruction and lineaging at cell resolution,” *Nature methods*, vol. 7, no. 7, pp. 547, 2010.
- [12] Kishore R Mosaliganti, Ramil R Noche, Fengzhu Xiong, Ian A Swinburne, and Sean G Megason, “Acme: automated cell morphology extractor for comprehensive reconstruction of cell membranes,” *PLoS computational biology*, vol. 8, no. 12, pp. e1002780, 2012.
- [13] Willis L. Jonsson, S. H. and Y. Refahi, “Research data supporting cell size and growth regulation in the arabidopsis thaliana apical stem cell niche,” *[Dataset]*, 2017.
- [14] Stanley R Sternberg, “Biomedical image processing,” *Computer*, , no. 1, pp. 22–34, 1983.
- [15] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein, “Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge,” in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 287–297.
- [16] Yuxin Wu and Kaiming He, “Group normalization,” *arXiv preprint arXiv:1803.08494*, 2018.
- [17] Po-Yu Kao, Thuyen Ngo, Angela Zhang, Jefferson Chen, and BS Manjunath, “Brain tumor segmentation and tractographic feature extraction from structural mr images for overall survival prediction,” *arXiv preprint arXiv:1807.07716*, 2018.
- [18] Pierre Soille, *Morphological image analysis: principles and applications*, Springer Science & Business Media, 2013.
- [19] Philipp Krähenbühl and Vladlen Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *Advances in neural information processing systems*, 2011, pp. 109–117.