Learning to Model Pixel-Embedded Affinity for Homogeneous Instance Segmentation

Wei Huang[1], **Shiyu Deng**[1], **Chang Chen**[1], **Xueyang Fu**[1], **Zhiwei Xiong**[1,2,*]

[1]University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

Abstract

Homogeneous instance segmentation aims to identify each instance in an image where all interested instances belong to the same category, such as plant leaves and microscopic cells. Recently, proposal-free methods, which straightforwardly generate instance-aware information to group pixels into different instances, have received increasing attention due to their efficient pipeline. However, they often fail to distinguish adjacent instances due to similar appearances, dense distribution and ambiguous boundaries of instances in homogeneous images. In this paper, we propose a pixel-embedded affinity modeling method for homogeneous instance segmentation, which is able to preserve the semantic information of instances and improve the distinguishability of adjacent instances. Instead of predicting affinity directly, we propose a self-correlation module to explicitly model the pairwise relationships between pixels, by estimating the similarity between embeddings generated from the input image through CNNs. Based on the self-correlation module, we further design a cross-correlation module to maintain the semantic consistency between instances. Specifically, we map the transformed input images with different views and appearances into the same embedding space, and then mutually estimate the pairwise relationships of embeddings generated from the original input and its transformed variants. In addition, to integrate the global instance information, we introduce an embedding pyramid module to model affinity on different scales. Extensive experiments demonstrate the versatile and superior performance of our method on three representative datasets. Code and models are available at https://github.com/weih527/Pixel-Embedded-Affinity.

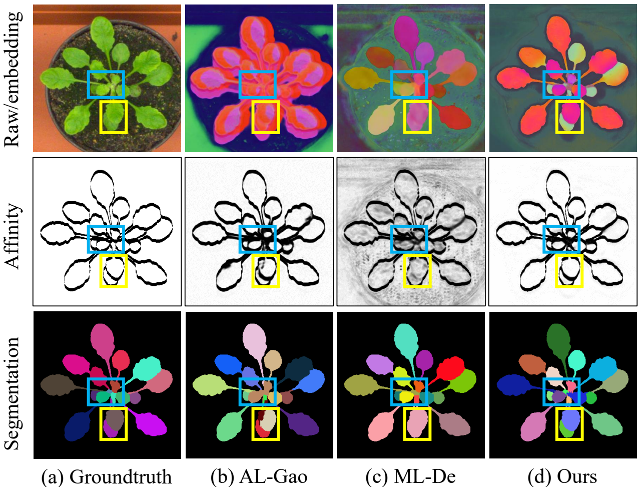

Figure 1: A visual example of representative methods for homogeneous instance segmentation on the CVPPP A1 dataset. Affinity learning (AL-Gao) straightforwardly generates affinity but suffers from the absence of semantic instance information. Metric learning (ML-De) aims to push all instances away from each other but ignores the spatial information between instances. Our method better preserves the semantic information of instances and pays more attention to the distinguishability of adjacent instances, producing superior segmentation results.

Introduction

Homogeneous instance segmentation, also referred to as intra-class segmentation sometimes, identifies all interested instances that belong to the same category in the target image. It has a wide range of applications, such as the phenotype measurement of cells (Yi et al. 2020) and plants (Scharr et al. 2016), the spatial arrangement of cancer nuclei (Yang et al. 2021) and the 3D reconstruction of neurons and mitochondria in connectomics (Funke et al. 2019; Wu et al. 2021; Li et al. 2022). Due to the complex characteristics of homogeneous images, such as similar appearances, dense distribution and ambiguous boundaries of instances, this challenging task attracts sustained attention, especially for biomedical images (Wu et al. 2018; Yi et al. 2020, 2021; Liu et al. 2021; Dong et al. 2019, 2020). With powerful feature representation capabilities, deep learning-based methods have become the mainstream, and remarkable progress has been made in instance segmentation (Gu, Deng, and Wei 2021; Hsieh et al. 2021; Vu, Kang, and Yoo 2021; Zhang et al. 2021). According to whether proposals are required, these methods can be divided into two categories, *i.e.*, proposal-based (He et al. 2017; Bolya et al. 2019) and proposal-free (Gao et al. 2019; Xie et al. 2020).

[*] Corresponding author: zwxiong@ustc.edu.cn

Recently, proposal-free methods have received increasing attention due to their efficient pipeline for the dense distribution of instances. These methods leverage deep convolutional neural networks (CNNs) to generate instance-aware information to group pixels into different instances by using clustering as post-processing.

Two representative kinds of proposal-free methods are affinity learning (Liu et al. 2018; Gao et al. 2019) and metric learning (De Brabandere et al. 2017; Lahoud et al. 2019). Affinity learning leverages CNNs to implicitly learn the pairwise relationships between pixels, *i.e.*, affinity. However, this implicit learning suffers from the absence of semantic instance information. As shown in Figure 1 (b), the embedding predicted by CNNs has limited semantic information. Metric learning adopts the discriminative loss to pull pixels belonging to the same instance together and push those of different instances away from each other in the embedding space. However, the spatial information of instances is ignored. In other words, it is easy to distinguish instances that are far apart in space but difficult to distinguish adjacent instances due to their similar appearances, dense distribution and ambiguous boundaries, as shown in Figure 1 (c). Therefore, in spite of the promising progress, homogeneous instance segmentation is still a challenging task, and there remains a large room for improvement.

In this paper, we propose a pixel-embedded affinity modeling method for homogeneous instance segmentation, which aims to preserve the semantic information of instances and improve the distinguishability of adjacent instances, as shown in Figure 1 (d). Different from affinity learning that predicts affinity directly, we propose a self-correlation module for explicit affinity modeling, which estimates the similarity between embeddings generated from the input image through a CNN. Moreover, we utilize affinity labels to supervise the learning of embeddings, which is distinct from metric learning that simply clusters embeddings belonging to the same instance together. Nevertheless, for the adjacent embeddings belonging to different instances, their corresponding receptive fields on the input image exist a large overlap. Thus, the adjacent embeddings are prone to be similar, which could produce incorrect affinity. To address this issue, we further design a cross-correlation module to improve the distinguishability of adjacent instances. Specifically, we construct transformed images with different views and appearances from the original input, and map them into an embedding space by a weight-sharing CNN of that used in the self-correlation module. By mutually estimating the pairwise relationships of embeddings generated from the input image and its transformed variants, the semantic consistency between instances can be better maintained. In addition, to integrate the global instance information, we introduce an embedding pyramid module to model affinity on different scales. It leverages the proposed self-correlation module to estimate the similarity of embeddings generated from different feature levels of the CNN.

Contributions of this paper are summarized as follows:

- We propose a pixel-embedded affinity modeling method for homogeneous instance segmentation. Our proposed self-correlation module explicitly models the pairwise relationship between pixels, which preserves the semantic instance information.
- We design a cross-correlation module by mutually estimating the pairwise relationships under different views and appearances of the input image to improve the distinguishability of adjacent instances.
- We introduce an embedding pyramid module by modeling affinity on different scales to integrate the global instance information.
- Extensive experiments on three representative datasets demonstrate the versatile and superior performance of our method for homogeneous instance segmentation.

Related Work

Instance segmentation is a fundamental task in computer vision (He et al. 2017; Hsieh et al. 2021; Vu, Kang, and Yoo 2021; Zhang et al. 2021). It requires not only classifying the category of each pixel correctly in an image, but also distinguishing each instance belonging to the same category at the same time. In contrast, homogeneous instance segmentation focuses on the identification of instances belonging to the same category in an image, which is desired in many practical applications, especially for biomedical image analysis (Funke et al. 2019; Chen, Strauch, and Merhof 2019; Yi et al. 2020, 2021; Yang et al. 2021; Liu et al. 2021).

Proposal-Based Instance Segmentation

Proposal-based methods combine object detection and segmentation, which first localize instances using bounding boxes and then segment instances within the cropped region-of-interest patches. As a fundamental work in this direction, Mask R-CNN (He et al. 2017) incorporates a mask branch into the region proposal network (Ren et al. 2017) to obtain instance masks from the predicted bounding boxes. Based on Mask R-CNN, many impressive works have been conducted for homogeneous instance segmentation (Yi et al. 2020, 2021; Yang et al. 2021; Liu et al. 2021). However, the performances of proposal-based methods are highly limited by the accuracy of the bounding boxes (Gao et al. 2019). Once the predictions of bounding boxes fail, the final instance masks are also incorrect. Especially, when instances are densely distributed in the target image, the bounding boxes of adjacent instances are easy to be suppressed due to the non-maximum suppression operation (Chen, Strauch, and Merhof 2019).

Proposal-Free Instance Segmentation

Recently, proposal-free methods have attracted more and more attention (Kirillov et al. 2017; Liu et al. 2018; Kong and Fowlkes 2018; Neven et al. 2019; Gao et al. 2019; Cheng et al. 2020). These methods not only shake off the drawback of proposals, but also operate faster than proposal-based methods. They first generate the instance-aware information, such as instance boundary (Kirillov et al. 2017), affinity (Liu et al. 2018) and embeddings (De Brabandere et al. 2017). Clustering algorithms are then adopted as post-processing to group pixels into different instances (Keuper et al. 2015; Fukunaga and Hostetler 1975).

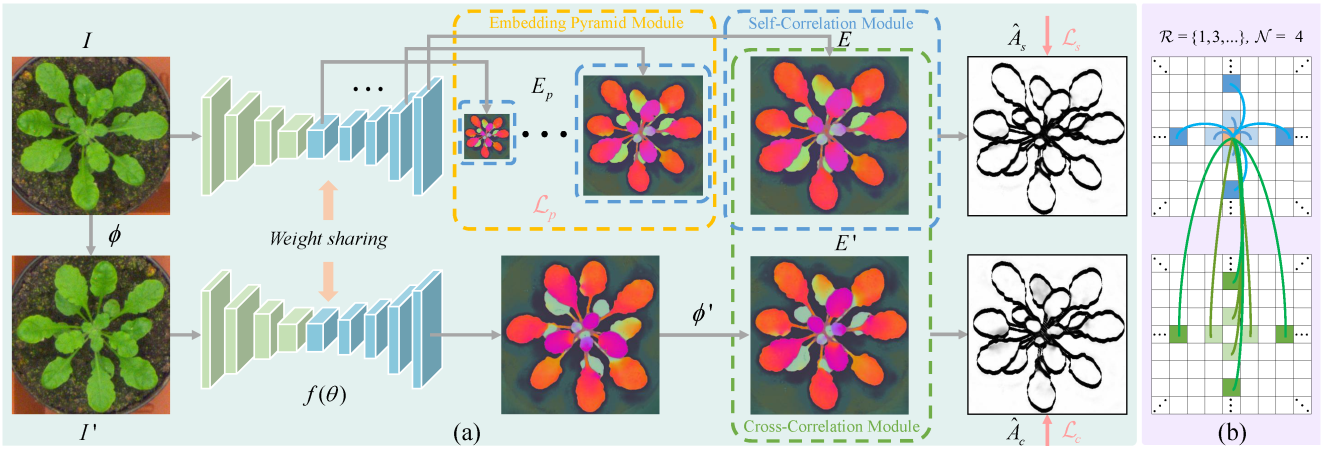

Figure 2: (a) The framework of our pixel-embedded affinity modeling method for homogeneous instance segmentation. It consists of three components, *i.e.*, Self-Correlation Module (SCM), Cross-Correlation Module (CCM) and Embedding Pyramid Module (EPM). The transformed image I' is generated from the original input I by the transformation function ϕ, *i.e.*, rotation, flipping, intensity adjustment and cutout. I and I' are then mapped into the embedding space E and E' by a CNN $f(\theta)$. SCM estimates the pairwise relationships between adjacent embeddings on E to generate the self-correlation affinity $\hat{A}_s$, while CCM mutually estimates the pairwise relationships of embeddings between E and E' to obtain the cross-correlation affinity $\hat{A}_c$. EPM models the relationships of embeddings on different scales of E (E_p) based on SCM. ϕ' is the inverse transformation of ϕ. $\mathcal{L}_s$, $\mathcal{L}_c$ and $\mathcal{L}_p$ are three loss functions to supervise the learning of E, E' and E_p, respectively. (b) The self/cross-correlation between the current embedding and its adjacent embeddings in a specified range $\mathcal{R}$ and neighborhood $\mathcal{N}$.

Affinity Learning. Affinity learning views the affinity as a multi-channel binary map and straightforwardly generates it by CNNs (Maire, Narihira, and Yu 2016; Tu et al. 2018; Liu et al. 2018; Gao et al. 2019; Lin et al. 2020; Wolf et al. 2020b; Xu et al. 2020). For example, GMIS (Liu et al. 2018) predicts the semantic map and the pixel affinity simultaneously to segment images. SSAP (Gao et al. 2019) and DaffNet (Xu et al. 2020) generate the affinity pyramid and then perform cascaded graph partition to obtain instance masks. However, these methods do not explicitly model the pairwise relationships between pixels and suffer from the absence of semantic instance information. This motivates us to explore an explicit way to model the affinity that better preserves the semantic information of instances.

Metric Learning. Metric learning leverages the discriminative loss to generate instance-aware embeddings, which aims to pull pixels belonging to the same instance together and push those of different instances away from each other (De Brabandere et al. 2017; Fathi et al. 2017; Kong and Fowlkes 2018; Payer et al. 2018; Konopczyński et al. 2018; Lahoud et al. 2019). However, since the relationship between all instances in the image need to be considered, these methods are often high computational complexity and are fragile to distinguish adjacent instances with similar appearances, dense distribution and ambiguous boundaries. To pay more attention to adjacent instances, Chen *et al.* (Chen, Strauch, and Merhof 2019) impose local constraints to only push adjacent instances far away instead of all instances. In contrast, we adopt local constraints between pixels rather than between instances. Moreover, we leverage affinity labels as strong supervision to guide the learning of embeddings. In other words, our method combines the advantages of affinity learning and metric learning.

Pixel-Embedded Affinity Modeling

In this section, we first briefly revisit the definition of affinity, which inspires the concept of pixel-embedded affinity modeling. The framework of our proposed method is shown in Figure 2, according to which we then describe the main components: self-correlation module (SCM), cross-correlation module (CCM) and embedding pyramid module (EPM) in detail.

Affinity Definition

For a given image $I \in \mathbb{R}^{H \times W}$, where H and W represent the image height and width, we aim to obtain its corresponding affinity $\hat{A} = [\hat{a}_1, \hat{a}_2, ..., \hat{a}_N] \in \mathbb{R}^{N \times H \times W}$ to describe the relationships between the current pixel and its adjacent pixels, where N represents the number of affinity channels. Specifically, each channel of affinity $\hat{a}_n (n = 1, 2, ..., N)$ describes the relationship between the current pixel and its n^{th} adjacent pixel. We can also obtain the affinity label $A = [a_1, a_2, ..., a_N]$ from the segmentation groundtruth $y \in \mathbb{R}^{H \times W}$ as

$$a_{n,i} = \begin{cases} 0, & \text{if } y_i \neq y_{i+n} \\ 1, & \text{if } y_i = y_{i+n}, \end{cases} \quad (1)$$

where the index i denotes the i^{th} pixel in the image I, and y_i is the segmentation ID of instance on this pixel. 0 means that the paired pixels i and $i+n$ belong to different instances in y, while 1 means the opposite.

SCM: Modeling Affinity Explicitly

Assuming that the relationships between pixels can be learned as the effective receptive field increases with deeper network layers (Ke et al. 2018), affinity learning directly predicts $\hat{A}$ by CNNs, and then adopts the mean square error (MSE) loss to penalize pixel-wise predictions independently. While strong supervision can be guaranteed, this implicit learning skips the definition of affinity, which leads to the absence of semantic instance information.

To explicitly estimate the pairwise relationships between pixels, we leverage a CNN $f(\theta)$ to map the input image I into an embedding space $E = f(I; \theta) \in \mathbb{R}^{D\times H\times W}$, where D denotes the dimension of each embedding vector that is a high-dimensional feature representation for the corresponding pixel in the image I. Inspired by metric learning, we assume embeddings of the same instance should be similar, while those of different instances should not. We adopt the cosine distance to measure the similarity between embeddings. The output of the cosine distance ranges from -1 to 1. 1 means that the two embeddings are exactly the same, while 0 means that they are orthogonal, *i.e.*, completely different. This is consistent with the definition of affinity in Eq. (1). Therefore, we define the self-correlation affinity $\hat{A}_s = [\hat{a}^s_1, \hat{a}^s_2, ..., \hat{a}^s_N]$ as the cosine distance between two embeddings

$$\hat{a}^s_{n,i} = cos(e_i, e_{i+n}) = \frac{e_i^T e_{i+n}}{||e_i||_2||e_{i+n}||_2}, \tag{2}$$

where e_i and $e_{i+n} \in \mathbb{R}^D$ are embeddings of pixel i and $i+n$ in E.

Metric learning often uses clustering to distinguish different instances, and the discriminative loss (De Brabandere et al. 2017) is widely utilized to penalize the similarity of all instances in an image. However, this global constraint ignores the spatial information of instances. As a rescue, we turn to affinity that represents local correlation of pixels, and adopt the MSE loss to supervise the learning of $\hat{A}_s$ as

$$\begin{aligned}\mathcal{L}_s &= ||\hat{A}_s - A||_2 \\ &= \frac{1}{N\times H\times W}\sum_{n=1}^{N}\sum_{i=1}^{H\times W}||\hat{a}^s_{n,i} - a_{n,i}||_2.\end{aligned} \tag{3}$$

In theory, we can consider the relationships between the current pixel and all other pixels in the image I, *i.e.*, the number of affinity channels $N = (H\times W - 1)/2$. However, besides the high computational complexity and the cost of GPU memories, the excessive range of affinity severely hinders the semantic information of instances, which is not beneficial for the learning of affinity (Gao et al. 2019). As shown in Figure 2 (b), typically, we only consider a set of specified ranges $\mathcal{R} = \{1, 3, 5, 9, 27\}$ in a 4-neighborhood (*i.e.*, $\mathcal{N} = 4$) during the training phase. In the inference phase, however, we can readily adjust the range and neighborhood to achieve adaptive affinity.

CCM: Distinguishing Adjacent Instances

Due to the large overlap of receptive fields between adjacent embeddings, it is still difficult to generate discriminative embeddings to distinguish adjacent instances. This obstacle motivates us to extract discriminative embeddings with semantic consistency from the input image under different conditions. To this end, we would like to find an effective transformation function ϕ to transform the input image I into its variant $I' = \phi(I)$, which enables the learning of discriminative embeddings.

We consider the transformed variant from two perspectives, *i.e.*, different views and different appearances for the same input. In general, convolutions are not transformation (*i.e.*, flipping, rotation) equivariant, meaning that if one rotates or flips the input, then the feature map will not rotate in a meaningful or easy-to-predict manner (Worrall et al. 2017). Leveraging this characteristic of CNNs, we apply flipping and rotation operations to map I to different views. To obtain different appearances of I, we randomly adjust the brightness and contrast of I (*i.e.*, intensity) and drop out parts of I (*i.e.*, cutout).

As shown in Figure 2 (a), we adopt a siamese structure of $f(\theta)$ to generate two embedding maps E and $E' = f(I'; \theta)$ from the original input I and its transformed counterparts I', simultaneously. We then mutually estimate the pairwise relationships between E and E' and obtain the cross-correlation affinity $\hat{A}_c = [\hat{a}^c_1, \hat{a}^c_2, ..., \hat{a}^c_N]$ as

$$\hat{a}^c_{n,i} = cos(e_i, e'_{i+n}), \tag{4}$$

which is also supervised by the affinity label A as

$$\mathcal{L}_c = ||\hat{A}_c - A||_2. \tag{5}$$

As shown in Figure 5, CCM effectively improves the distinguishability of adjacent instances.

EPM: Integrating Global Information

Large-scale and small-scale instances often co-exist in the target image. It is thus difficult to find a suitable affinity range set $\mathcal{R}$ to model their correlations at the same time, since long-range affinity hinders the semantic information of small instances while short-range affinity cannot model the correlations of large instances well. Inspired by (Gao et al. 2019), we address this problem by multi-scale affinity modeling, *i.e.*, small instances are considered at a higher resolution while large instances at a lower resolution.

As shown in Figure 2 (a), we generate embeddings E_p with different resolutions on different scales of the decoder of $f(\theta)$, where E_p denotes the embeddings at the p^{th} layer of decoder. Dependent on the structure of decoder, we set the maximal p to 4. In other words, embeddings are predicted under $\{1/2, 1/4, 1/8, 1/16\}$ resolutions of the original input I. Leveraging Eq. (2), The corresponding affinity $\hat{A}_p$ is obtained by estimating self-correlation on the predicted embedding E_p. We further supervise the affinity on different scales as

$$\mathcal{L}_p = \sum_p ||\hat{A}_p - A_p||_2, \tag{6}$$

where A_p represents the affinity label generated from the correspondingly downsampled segmentation groundtruth y_p. Due to the change of instance size, we gradually shrink the range set of affinity as the resolution decreases. As shown in Figure 5, EPM effectively avoids the merging of large-scale and small-scale instances.

Methods	Param.	SBD	$\vert DiC\vert$
MSU (Scharr et al. 2016)	-	66.7	2.3
Nottingham (Scharr et al. 2016)	-	68.3	3.8
Wageningen (Yin et al. 2014)	-	71.1	2.2
IPK (Pape and Klukas 2014)	-	74.4	2.6
Coloring (Kulikov et al. 2018)	30.2M	80.4	2.0
ML-De (De Brabandere et al. 2017)	23.1M	84.2	1.0
Recurrent (Ren and Zemel 2017)	-	84.9	**0.8**
Aug. (Kuznichov et al. 2019)	-	88.7	5.3
Harmonic (Kulikov et al. 2020)	43.1M	89.0	3.0
Synthesis (Ward et al. 2018)	105.7M	90.0	-
PFFNet (Liu et al. 2021)	105.7M	91.1	-
Ours w/ ResNet-50	15.3M	91.7	1.5
Ours w/ ResNet-101	34.3M	91.9	1.4
Ours w/ ResUNet	**4.7M**	**92.3**	2.4

Table 1: Quantitative comparison with state-of-the-art methods on the test set of CVPPP A1.

Training and Inference Details

Our affinity modeling method is network-agnostic. That is to say, the CNN backbone in the framework can be replaced by arbitrary advanced structures. The above three loss functions are combined for the end-to-end training as

$$\mathcal{L}_{total} = \alpha\mathcal{L}_s + \beta\mathcal{L}_c + \gamma\mathcal{L}_p, \tag{7}$$

where α, β and γ are weighting coefficients to balance these three terms.

In the inference phase, we only use the self-correlation affinity $\hat{A}_s$ from Eq. (3) as the final affinity prediction, and we adopt the Mutex algorithm (Wolf et al. 2020a) as post-processing to obtain instance masks from $\hat{A}_s$. In addition, we merge too small instances to further refine the final segmentation result.

Experiments

Datasets

CVPPP. The Computer Vision Problems in Plant Phenotyping (CVPPP) dataset (Minervini et al. 2016) is one of the most popular benchmarks for homogeneous instance segmentation, where the task is to segment individual leaf instances of a plant growing in a pot. The dataset consists of five subsets of different plants. Following (Ren and Zemel 2017; Kulikov et al. 2020; Liu et al. 2021), we adopt the most commonly used subset A1 to demonstrate the superiority of our proposed method. This subset consists of 128 training images (530 × 500) with public ground truth labels and 33 test images with no publicly available labels. We randomly select 20 images from the training set as the validation set. To evaluate the performance of our method, the predicted results of test images are submitted to the official evaluation platform. In order to demonstrate the generalizability of our method, we further adopt the subset A2 containing 31 images with publicly available labels as another test set. Two common metrics for quantitative evaluation, *i.e.*, symmetric best Dice (SBD) and absolute difference in counting ($|DiC|$).

Dataset	Methods	Clustering	SBD	$\vert DiC\vert$
A1	AL-Gao	Mutex	87.1	1.25
	ML-De	Mean-shift	87.3	1.45
	ML-De	Mutex	88.5	1.10
	Ours	Mutex	**89.1**	**0.85**
A2	AL-Gao	Mutex	71.1	2.61
	ML-De	Mean-shift	71.2	2.52
	ML-De	Mutex	73.4	2.00
	Ours	Mutex	**76.3**	**1.71**

Table 2: Quantitative comparison on the validation sets of CVPPP A1 and A2.

BBBC039V1. The BBBC039V1 dataset from (Ljosa, Sokolnicki, and Carpenter 2012) contains 200 images (520× 696) obtained from fluorescence microscopy (FM). The main use of this dataset is the study of segmentation algorithms that can separate individual nuclei instances accurately, regardless of their shape and cell density. Following the official data split, we use 100 images for training, 50 for validation and the rest 50 for testing. Three common metrics for cell segmentation in FM images are reported for performance evaluation, *i.e.*, Aggregated Jaccard Index (AJI), pixel-level Dice score ($Dice$) and Panoptic Quality (PQ).

AC3/AC4. To demonstrate the effectiveness of our proposed method on the 3D instance segmentation task, we further conduct experiments on a common electron microscope (EM) dataset, *i.e.*, AC3/AC4. This task treats an individual neuron as one instance and aims to reconstruct each 3D neuron in the given volume (2D image sequences). AC3/AC4 are two labeled subsets cropped from the mouse somatosensory cortex dataset of (Kasthuri et al. 2015) acquired at $3 \times 3 \times 29\ nm^3$ resolution. They contain 256 and 100 sequential images (1024 × 1024), respectively. Following the SNEMI3D challenge (Arganda-Carreras et al. 2015), we adopt the top 80 sections of AC4 as the training set, the remaining 20 sections as the validation set and the top 100 sections of AC3 as the test set. Two widely used metrics for neuron segmentation in EM images are reported for performance evaluation, *i.e.*, variation of information ($VOI = VOI_S + VOI_M$) and adapted Rand error ($ARAND$), where VOI_S and VOI_M represent split errors and merge errors, respectively.

Implementation Details

We adopt three representative CNN backbones to demonstrate the superiority of our method on the CVPPP dataset, *i.e.*, residual U-Net (ResUNet) (Ronneberger, Fischer, and Brox 2015) and ResNet-50/101 (He et al. 2016). The best performer, ResUNet, is then used as the backbone on the BBBC039V1 dataset. For the 3D EM dataset, we adopt 3D ResUNet (Lee et al. 2017) as the backbone. Following (Kulikov et al. 2020), we adopt the basic augmentations (flipping, cropping and scaling) to extend the training set of CVPPP and BBBC039V1, and add the elastic augmentation for AC3/AC4 (Funke et al. 2019; Huang et al. 2020).

Methods	AJI	$Dice$	PQ
Mask R-CNN (He et al. 2017)	0.7983	0.9277	0.7773
Cell R-CNN (Zhang et al. 2018)	0.8070	0.9290	0.7959
UPSNet (Xiong et al. 2019)	0.8128	0.9274	0.7857
JSISNet (De Geus et al. 2018)	0.8134	0.9316	0.7913
PanFPN (Kirillov et al. 2019)	0.8193	0.9320	0.7960
OANet (Liu et al. 2019b)	0.8198	0.9372	0.8085
AUNet (Li et al. 2019)	0.8252	0.9377	0.8090
Cell R-CNN v2 (Liu et al. 2019a)	0.8260	0.9336	0.8010
PFFNet (Liu et al. 2021)	0.8477	0.9478	0.8330
Ours	**0.8674**	**0.9673**	**0.8420**

Table 3: Quantitative comparison with state-of-the-art methods on the test set of BBBC039V1.

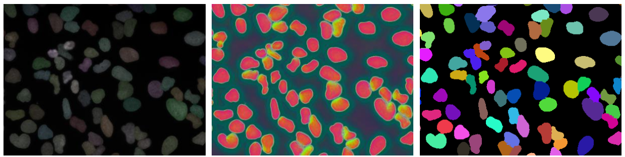

Figure 3: Visual results on the test set of BBBC039V1.

We train these networks using Adam (Kingma and Ba 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 0.0001, and a batch size of 2 on an *NVIDIA Titan XP* GPU for $200,000$ iterations. The weighting coefficients of the loss function is empirically set as $\alpha = \beta = \gamma = 1$. The number of embedding dimensions is set to 16.

Results

Results on CVPPP. We first compare our method with the state-of-the-art methods on the test set of CVPPP A1. As listed in Table 1, our method currently achieves the best result on the main SBD metric. Compared with the most recent methods, our method also obtains superior performance on the $|DiC|$ metric with significantly fewer parameters. We then compare our method with two most relevant methods (*i.e.*, affinity learning (AL-Gao) (Gao et al. 2019) and metric learning (ML-De) (De Brabandere et al. 2017)) with the same CNN backbone (ResUNet). For ML-De, its default post-processing is the mean-shift clustering algorithm. For a fair comparison, we use our proposed SCM to convert the predicted embeddings into affinity, and then use the same post-processing (Mutex) of AL-Gao and Ours to obtain final instance masks. As listed in Table 2, our method outperforms both AL-Gao and ML-De on the validation sets of A1 and A2. Especially, when trained on A1 and tested on A2, our method significantly outperforms the competitors on the SBD and $|DiC|$ metric, which demonstrates its superior generalizability.

We further qualitatively compare our method with AL-Gao and ML-De, as shown in Figure 1. Since AL-Gao does not output embeddings, we adopt the feature of the penultimate layer of the network as its embeddings. From the visual results, we have three observations: (1) Since AL-Gao views the prediction of affinity as a binary output, its embeddings do not have meaningful semantic information, as can be seen from the first row (embedding). (2) ML-De aims to push all instances far away from each other, but ignores the spatial information between instances, so adjacent instances are difficult to distinguish, as can be seen from the second row (affinity). (3) Compared with AL-Gao, our method better preserves the semantic information of instances. Different from ML-De, our method pays more attention to adjacent instances. Both promote segmentation accuracy, as can be seen in the third row (marked boxes).

Methods	VOI_S	VOI_M	VOI	$ARAND$
ML-De	1.5752	0.6151	2.1903	0.1964
SuperHuman	1.1445	0.2630	1.4075	0.1220
MALA	1.3039	0.2423	1.5462	0.1203
Ours	**0.8522**	**0.2322**	**1.0844**	**0.0938**

Table 4: Quantitative comparison with metric learning (ML-De (De Brabandere et al. 2017)) and two affinity learning methods (*i.e.*, SuperHuman (Lee et al. 2017) and MALA (Funke et al. 2019)) on the test set of AC3/AC4.

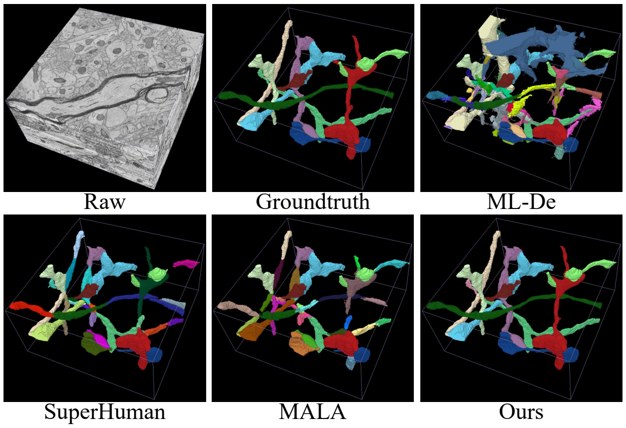

Figure 4: Visual comparison on the test set of AC3/AC4. We select 10 neurons for qualitative comparison.

Results on BBBC039V1. When employing our method on the BBBC039V1 dataset, we adjust the range set of affinity to $[1, 3, 5, 9, 11]$, since the size of cells is relatively small. As listed in Table 3, our method outperforms existing methods on this dataset by a large margin. Visual results in Figure 3 demonstrate that our method successfully distinguishes adjacent cells, which is the main challenge of this dataset.

Results on AC3/AC4. We compare our method with two popular methods for neuron segmentation, *i.e.*, SuperHuman (Lee et al. 2017) and MALA (Funke et al. 2019), both of which belong to affinity learning. In addition, we reproduce ML-De (De Brabandere et al. 2017) on this task to compare with metric learning. For a fair comparison, we adopt the same backbone (3D ResUNet) (Lee et al. 2017) and the same post-processing (the Multicut algorithm (Beier et al. 2017)).

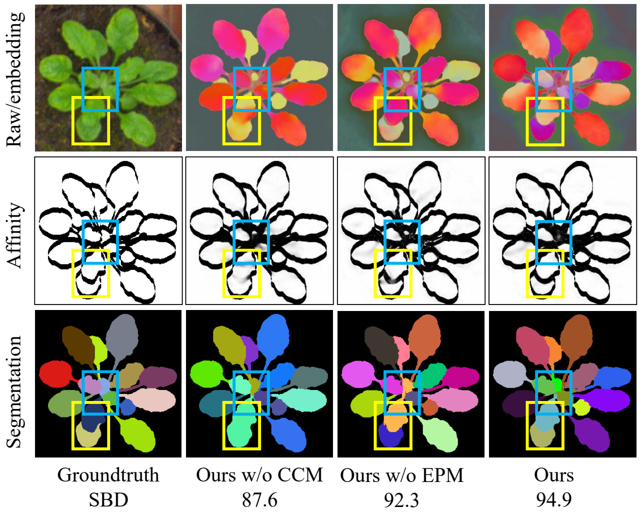

Figure 5: Visual demonstration for the effectiveness of modules on the CVPPP A1 dataset.

SCM	CCM	EPM	SBD	$\|DiC\|$
✓			87.7	1.15
✓		✓	88.1	1.00
✓	✓		88.5	0.95
✓	✓	✓	**89.1**	**0.85**

Table 5: Ablation results for the effectiveness of modules on the CVPPP A1 dataset.

Table 4 demonstrates the superiority of our method over its competitors. As shown in Figure 4, our method effectively avoids split and merge errors.

Ablation Studies

We conduct comprehensive ablation studies with ResUNet on the validation set of the CVPPP dataset.

The Effectiveness of Modules. As listed in Table 5, our method achieves the best performance when three modules are adopted at the same time. We further qualitatively demonstrate their effectiveness, as shown in Figure 5. When CCM is removed, the distinguishability of adjacent instances is reduced. Due to the absence of global instance information, large-scale and small-scale instances are prone to be merged together when EPM is removed.

The Effectiveness of Transformations. Table 6 demonstrates the effectiveness of each selected transformation. Note that, once the flipping and rotation transformation (Flip. & Rot.) is removed, the segmentation performance drops dramatically, which demonstrates that different views can effectively reduce the overlap of receptive fields and improve the distinguishability of adjacent instances.

Dimension of Embeddings. As shown in Figure 6 (a), higher dimensional embeddings are more beneficial for the representation of pixels in the image.

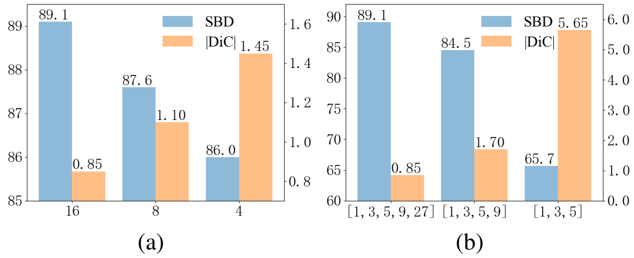

Figure 6: Ablation results for dimension of embeddings (a) and range set of affinity (b) on the CVPPP A1 dataset.

Flip. & Rot.	Cutout	Intensity	SBD	$\|DiC\|$
	✓	✓	86.5	1.35
✓		✓	87.7	1.00
✓	✓		88.6	**0.85**
✓	✓	✓	**89.1**	**0.85**

Table 6: Ablation results for the effectiveness of transformations on the CVPPP A1 dataset.

$\mathcal{R}$	$\mathcal{N}$	SBD	$\|DiC\|$
$[1, 3, 5, 9, 27]$	4	76.3	1.71
$[1, 3, 5, 9, 27]$	8	77.1	1.68
$[1, 3, 5, 7, 9, 11, 19, 27, 35]$	8	**77.4**	**1.55**

Table 7: Ablation results for adaptive affinity by extending the ranges and neighborhoods of affinity during inference on the CVPPP A2 dataset.

Range Set of Affinity. Figure 6 (b) demonstrates that, as the range set of affinity shrinks, the segmentation performance drops. It verifies that choosing a reasonably large affinity range set is beneficial to the segmentation results.

Adaptive Affinity. Although we use fixed ranges and neighborhoods of affinity for training, we can generate affinity with arbitrary ranges and neighborhoods from the learned model in the inference phase. As listed in Table 7, extending the ranges and neighborhoods of affinity during inference can further improve the segmentation performance.

Conclusion

In this paper, we propose a pixel-embedded affinity modeling method for homogeneous instance segmentation. By jointly taking advantage of affinity learning and metric learning, the proposed self-correlation module enables explicit affinity modeling. The cross-correlation module is further designed to improve the distinguishability of adjacent instances. In addition, the embedding pyramid module is introduced to integrate the global instance information by modeling affinity at different scales. Through evaluation on three representative datasets, we demonstrate the versatile and superior performance of our proposed method.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2017YFA0700800, the National Natural Science Foundation of China under Grant 62021001, and the University Synergy Innovation Program of Anhui Province No. GXXT-2019-025.

References

Arganda-Carreras, I.; Turaga, S. C.; Berger, D. R.; Cireşan, D.; Giusti, A.; Gambardella, L. M.; Schmidhuber, J.; Laptev, D.; Dwivedi, S.; Buhmann, J. M.; et al. 2015. Crowdsourcing the Creation of Image Segmentation Algorithms for Connectomics. *Frontiers in Neuroanatomy*, 9: 142.

Beier, T.; Pape, C.; Rahaman, N.; Prange, T.; Berg, S.; Bock, D. D.; Cardona, A.; Knott, G. W.; Plaza, S. M.; Scheffer, L. K.; et al. 2017. Multicut Brings Automated Neurite Segmentation Closer to Human Performance. *Nature Methods*, 14(2): 101–102.

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. YOLACT: Real-Time Instance Segmentation. In *ICCV*.

Chen, L.; Strauch, M.; and Merhof, D. 2019. Instance Segmentation of Biomedical Images with an Object-Aware Embedding Learned with Local Constraints. In *MICCAI*.

Cheng, B.; Collins, M. D.; Zhu, Y.; Liu, T.; Huang, T. S.; Adam, H.; and Chen, L.-C. 2020. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *CVPR*.

De Brabandere, B.; et al. 2017. Semantic Instance Segmentation with a Discriminative Loss Function. *arXiv preprint arXiv:1708.02551*.

De Geus, D.; et al. 2018. Panoptic Segmentation with a Joint Semantic and Instance Segmentation Network. *arXiv preprint arXiv:1809.02110*.

Dong, M.; Liu, D.; Xiong, Z.; Chen, X.; Zhang, Y.; Zha, Z.-J.; Bi, G.; and Wu, F. 2019. Instance Segmentation from Volumetric Biomedical Images without Voxel-Wise Labeling. In *MICCAI*.

Dong, M.; Liu, D.; Xiong, Z.; Chen, X.; Zhang, Y.; Zha, Z.-J.; Bi, G.; and Wu, F. 2020. Towards Neuron Segmentation from Macaque Brain Images: A Weakly Supervised Approach. In *MICCAI*.

Fathi, A.; Wojna, Z.; Rathod, V.; Wang, P.; Song, H. O.; Guadarrama, S.; and Murphy, K. P. 2017. Semantic Instance Segmentation via Deep Metric Learning. *arXiv preprint arXiv:1703.10277*.

Fukunaga, K.; and Hostetler, L. 1975. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, 21(1): 32–40.

Funke, J.; Tschopp, F.; Grisaitis, W.; Sheridan, A.; Singh, C.; Saalfeld, S.; and Turaga, S. C. 2019. Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7): 1669–1680.

Gao, N.; Shan, Y.; Wang, Y.; Zhao, X.; Yu, Y.; Yang, M.; and Huang, K. 2019. SSAP: Single-Shot Instance Segmentation with Affinity Pyramid. In *ICCV*.

Gu, Y.; Deng, C.; and Wei, K. 2021. Class-Incremental Instance Segmentation via Multi-Teacher Networks. In *AAAI*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Hsieh, T.-I.; Robb, E.; Chen, H.-T.; and Huang, J.-B. 2021. DropLoss for Long-Tail Instance Segmentation. In *AAAI*.

Huang, W.; Chen, C.; Xiong, Z.; Zhang, Y.; Liu, D.; and Wu, F. 2020. Learning to Restore ssTEM Images from Deformation and Corruption. In *ECCVW*.

Kasthuri, N.; Hayworth, K. J.; Berger, D. R.; Schalek, R. L.; Conchello, J. A.; Knowles-Barley, S.; Lee, D.; Vázquez-Reina, A.; Kaynig, V.; Jones, T. R.; et al. 2015. Saturated Reconstruction of a Volume of Neocortex. *Cell*, 162(3): 648–661.

Ke, T.-W.; Hwang, J.-J.; Liu, Z.; and Yu, S. X. 2018. Adaptive Affinity Fields for Semantic Segmentation. In *ECCV*.

Keuper, M.; Levinkov, E.; Bonneel, N.; Lavoué, G.; Brox, T.; and Andres, B. 2015. Efficient Decomposition of Image and Mesh Graphs by Lifted Multicuts. In *ICCV*.

Kingma, D. P.; and Ba, J. L. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Kirillov, A.; Girshick, R.; He, K.; and Dollár, P. 2019. Panoptic Feature Pyramid Networks. In *CVPR*.

Kirillov, A.; Levinkov, E.; Andres, B.; Savchynskyy, B.; and Rother, C. 2017. InstanceCut: from Edges to Instances with MultiCut. In *CVPR*.

Kong, S.; and Fowlkes, C. C. 2018. Recurrent Pixel Embedding for Instance Grouping. In *CVPR*.

Konopczyński, T.; Kröger, T.; Zheng, L.; and Hesser, J. 2018. Instance Segmentation of Fibers from Low Resolution CT Scans via 3D Deep Embedding Learning. In *BMVC*.

Kulikov, V.; et al. 2018. Instance Segmentation by Deep Coloring. *arXiv preprint arXiv:1807.10007*.

Kulikov, V.; et al. 2020. Instance Segmentation of Biological Images Using Harmonic Embeddings. In *CVPR*.

Kuznichov, D.; Zvirin, A.; Honen, Y.; and Kimmel, R. 2019. Data Augmentation for Leaf Segmentation and Counting Tasks in Rosette Plants. In *CVPRW*.

Lahoud, J.; Ghanem, B.; Pollefeys, M.; and Oswald, M. R. 2019. 3D Instance Segmentation via Multi-Task Metric Learning. In *ICCV*.

Lee, K.; Zung, J.; Li, P.; Jain, V.; and Seung, H. S. 2017. Superhuman Accuracy on the SNEMI3D Connectomics Challenge. *arXiv preprint arXiv:1706.00120*.

Li, M.; Chen, C.; Liu, X.; Huang, W.; Zhang, Y.; and Xiong, Z. 2022. Advanced Deep Networks for 3D Mitochondria Instance Segmentation. In *ISBI*.

Li, Y.; Chen, X.; Zhu, Z.; Xie, L.; Huang, G.; Du, D.; and Wang, X. 2019. Attention-Guided Unified Network for Panoptic Segmentation. In *CVPR*.

Lin, F.; Li, B.; Zhou, W.; Li, H.; and Lu, Y. 2020. Single-Stage Instance Segmentation. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(3): 1–19.

Liu, D.; Zhang, D.; Song, Y.; Huang, H.; and Cai, W. 2021. Panoptic Feature Fusion Net: A Novel Instance Segmentation Paradigm for Biomedical and Biological Images. *IEEE Transactions on Image Processing*, 30: 2045–2059.

Liu, D.; Zhang, D.; Song, Y.; Zhang, C.; Zhang, F.; O'Donnell, L.; and Cai, W. 2019a. Nuclei Segmentation via a Deep Panoptic Model with Semantic Feature Fusion. In *IJCAI*.

Liu, H.; Peng, C.; Yu, C.; Wang, J.; Liu, X.; Yu, G.; and Jiang, W. 2019b. An End-to-End Network for Panoptic Segmentation. In *CVPR*.

Liu, Y.; Yang, S.; Li, B.; Zhou, W.; Xu, J.; Li, H.; and Lu, Y. 2018. Affinity Derivation and Graph Merge for Instance Segmentation. In *ECCV*.

Ljosa, V.; Sokolnicki, K. L.; and Carpenter, A. E. 2012. Annotated High-Throughput Microscopy Image Sets for Validation. *Nature Methods*, 9(7): 637–637.

Maire, M.; Narihira, T.; and Yu, S. X. 2016. Affinity CNN: Learning Pixel-Centric Pairwise Relations for Figure/Ground Embedding. In *CVPR*.

Minervini, M.; Fischbach, A.; Scharr, H.; and Tsaftaris, S. A. 2016. Finely-Grained Annotated Datasets for Image-Based Plant Phenotyping. *Pattern Recognition Letters*, 81: 80–89.

Neven, D.; Brabandere, B. D.; Proesmans, M.; and Gool, L. V. 2019. Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth. In *CVPR*.

Pape, J.-M.; and Klukas, C. 2014. 3-D Histogram-Based Segmentation and Leaf Detection for Rosette Plants. In *ECCVW*.

Payer, C.; Štern, D.; Neff, T.; Bischof, H.; and Urschler, M. 2018. Instance Segmentation and Tracking with Cosine Embeddings and Recurrent Hourglass Networks. In *MICCAI*.

Ren, M.; and Zemel, R. S. 2017. End-to-End Instance Segmentation with Recurrent Attention. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. volume 39, 1137–1149.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*.

Scharr, H.; Minervini, M.; French, A. P.; Klukas, C.; Kramer, D. M.; Liu, X.; Luengo, I.; Pape, J.-M.; Polder, G.; Vukadinovic, D.; et al. 2016. Leaf Segmentation in Plant Phenotyping: A Collation Study. *Machine Vision and Applications*, 27(4): 585–606.

Tu, W.-C.; Liu, M.-Y.; Jampani, V.; Sun, D.; Chien, S.-Y.; Yang, M.-H.; and Kautz, J. 2018. Learning Superpixels with Segmentation-Aware Affinity Loss. In *CVPR*.

Vu, T.; Kang, H.; and Yoo, C. D. 2021. SCNet: Training Inference Sample Consistency for Instance Segmentation. In *AAAI*.

Ward, D.; et al. 2018. Deep Leaf Segmentation Using Synthetic Data. In *BMVC*.

Wolf, S.; Bailoni, A.; Pape, C.; Rahaman, N.; Kreshuk, A.; Köthe, U.; and Hamprecht, F. A. 2020a. The Mutex Watershed and its Objective: Efficient, Parameter-Free Graph Partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3724–3738.

Wolf, S.; Li, Y.; Pape, C.; Bailoni, A.; Kreshuk, A.; and Hamprecht, F. A. 2020b. The Semantic Mutex Watershed for Efficient Bottom-Up Semantic Instance Segmentation. In *ECCV*.

Worrall, D. E.; Garbin, S. J.; Turmukhambetov, D.; and Brostow, G. J. 2017. Harmonic Networks: Deep Translation and Rotation Equivariance. In *CVPR*.

Wu, S.; Chen, C.; Xiong, Z.; Chen, X.; and Sun, X. 2021. Uncertainty-Aware Label Rectification for Domain Adaptive Mitochondria Segmentation. In *MICCAI*.

Wu, Z.; Chang, R.; Ma, J.; Lu, C.; and Tang, C.-K. 2018. Annotation-Free and One-Shot Learning for Instance Segmentation of Homogeneous Object Clusters. In *IJCAI*.

Xie, E.; Sun, P.; Song, X.; Wang, W.; Liu, X.; Liang, D.; Shen, C.; and Luo, P. 2020. PolarMask: Single Shot Instance Segmentation with Polar Representation. In *CVPR*.

Xiong, Y.; Liao, R.; Zhao, H.; Hu, R.; Bai, M.; Yumer, E.; and Urtasun, R. 2019. UPSNet: A Unified Panoptic Segmentation Network. In *CVPR*.

Xu, X.; Chiu, M. T.; Huang, T. S.; and Shi, H. 2020. Deep Affinity Net: Instance Segmentation via Affinity. *arXiv preprint arXiv:2003.06849*.

Yang, S.; Zhang, J.; Huang, J.; Lovell, B. C.; and Han, X. 2021. Minimizing Labeling Cost for Nuclei Instance Segmentation and Classification with Cross-Domain Images and Weak Labels. In *AAAI*.

Yi, J.; Tang, H.; Wu, P.; Liu, B.; Hoeppner, D. J.; Metaxas, D. N.; Han, L.; and Fan, W. 2020. Object-Guided Instance Segmentation for Biological Images. In *AAAI*.

Yi, J.; Wu, P.; Tang, H.; Liu, B.; Huang, Q.; Qu, H.; Han, L.; Fan, W.; Hoeppner, D. J.; and Metaxas, D. N. 2021. Object-Guided Instance Segmentation With Auxiliary Feature Refinement for Biological Images. *IEEE Transactions on Medical Imaging*, 40(9): 2403–2414.

Yin, X.; Liu, X.; Chen, J.; and Kramer, D. M. 2014. Multi-Leaf Tracking from Fluorescence Plant Videos. In *ICIP*.

Zhang, D.; Song, Y.; Liu, D.; Jia, H.; Liu, S.; Xia, Y.; Huang, H.; and Cai, W. 2018. Panoptic Segmentation with an End-to-End Cell R-CNN for Pathology Image Analysis. In *MICCAI*.

Zhang, G.; Lu, X.; Tan, J.; Li, J.; Zhang, Z.; Li, Q.; and Hu, X. 2021. RefineMask: Towards High-Quality Instance Segmentation with Fine-Grained Features. In *CVPR*.