Weakly Supervised Cell-Instance Segmentation with Two Types of Weak Labels by Single Instance Pasting

Kazuya Nishimura, Ryoma Bise
Kyushu University, Fukuoka, Japan
kazuya.nishimura@human.ait.kyushu-u.ac.jp

Abstract

Cell instance segmentation that recognizes each cell boundary is an important task in cell image analysis. While deep learning-based methods have shown promising performances with a certain amount of training data, most of them require full annotations that show the boundary of each cell. Generating the annotation for cell segmentation is time-consuming and human labor. To reduce the annotation cost, we propose a weakly supervised segmentation method using two types of weak labels (one for cell type and one for nuclei position). Unlike general images, these two labels are easily obtained in phase-contrast images. The intercellular boundary, which is necessary for cell instance segmentation, cannot be directly obtained from these two weak labels, so to generate the boundary information, we propose a single instance pasting based on the copy-and-paste technique. First, we locate single-cell regions by counting cells and store them in a pool. Then, we generate the intercellular boundary by pasting the stored single-cell regions to the original image. Finally, we train a boundary estimation network with the generated labels and perform instance segmentation with the network. Our evaluation on a public dataset demonstrated that the proposed method achieves the best performance among the several weakly supervised methods we compared.

1. Introduction

Phase-contrast microscopy is widely used for long-term monitoring of living cells without staining. Instance segmentation that recognizes each cell boundary in a phase-contrast image provides key information for cell morphological analysis and cell behavior analysis [10]. Since the phase-contrast image contains a large number of cells in one image (over one hundred), automated cell segmentation is required for large-scale analysis. As shown in Figure 1, the cell boundary is typically ambiguous, and the cell has various morphologies. Therefore, instance segmentation is a challenging task.

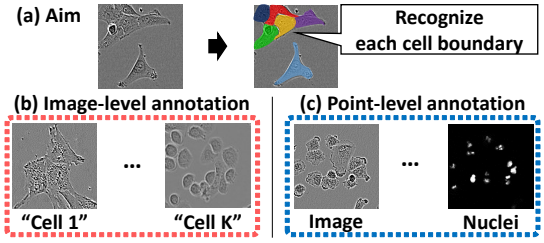

Figure 1. Weak supervision settings. (a) Our recognition aim. (b) Image-level annotation (cell type label). (c) Point-level annotation (cell position label).

Deep learning-based cell segmentation methods [31, 6, 27, 33, 10] have achieved promising results with a certain amount of training data. However, cell segmentation requires instance-level annotation indicating the boundaries of each cell for each imaging condition (*e.g.,* type of cell, microscopy, growth factor, and density). Collecting these annotations is time-consuming and human labor.

Weakly supervised segmentation, which performs segmentation with an easily obtained annotation rather than pixel-level annotation, is one promising solution to reduce annotation costs [17, 40, 42, 29, 5, 43, 26]. Image-level annotation (*i.e.,* a class label) is widely utilized as a weak label in general images [17, 2, 44] and organ images [40]. However, it is difficult to recognize the boundaries of the same class instance from the image-level annotation since it only contains semantic information. Point-level annotation, which indicates the cell position, is mostly used on cell or nuclei segmentation tasks [42, 29, 5, 43, 26]. While point-level annotation contains instance clues, there is no boundary information. Some methods use the color or contrast information of the input image to complement boundary information [42, 29]. However, the contrast of the phase-contrast image is typically low, and the foreground and background pixels tend to have a similar value, as shown in Figure 1(a). This makes it difficult to recognize individual cell boundaries in a phase-contrast image only using

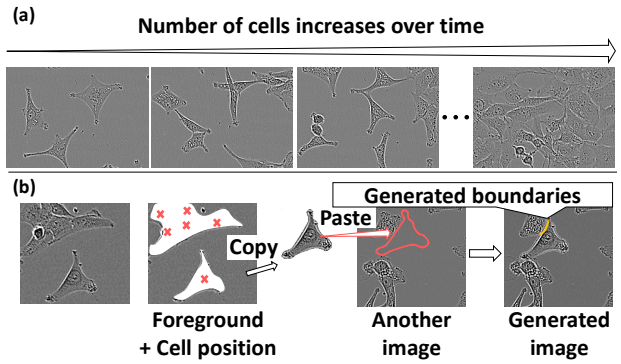

Figure 2. Key concept of proposed method. (a) Characteristics of phase-contrast image. (b) Our idea of single instance pasting. If cells are captured from seeding, there are single-cell regions in an initial state, and the number of cells gradually increases over time by cell division. Our idea with single instance pasting is to find a single cell region from the initial state and then paste the region into another image in order to generate intercellular boundaries.

the point-level information. It is therefore necessary to utilize additional boundary information to identify each cell boundary.

The base idea of this paper is to use two types of weak labels, namely, image-level annotation and point-level annotation, for complementing the lack of information. In the phase-contrast image, these labels can be easily obtained without additional manual annotation costs. The first type is a cell-type label (Figure 1(b)). Usually, a single cell type is used for experiments when observing cells, and the type of cells used is recorded, which means the cell class label can be automatically collected without additional costs. We can obtain foreground information from the image-level annotation by using an off-the-shelf CAM-based weakly supervised segmentation method [17]. The second type is a cell nuclei position label, which can be automatically obtained from nuclei-stained cells by simultaneously capturing phase-contrast and fluorescent images (Figure 1(c)). These labels can also be collected without additional human annotation costs. By using these two labels, we can obtain information on foreground regions and cell positions.

In this paper, we propose a weakly supervised instance segmentation method using the cell type label and cell position label. The foreground region and the cell position are obtained from these two weak labels, but intercellular boundaries are not provided. We, therefore, generate labels that include intercellular boundary information ourselves using a newly developed single instance pasting. Finally, we train a boundary estimation network with the self-generated labels and perform instance segmentation.

The objective of the single instance pasting is to locate single-cell instances and paste them into another image. As shown in Figure 2(a), the cell density gradually increases over time by cell division. Thus, the single-cell instance is easily obtained from the initial state of time-lapse images by counting cells using the foreground regions and cell positions that are obtained from the two weak labels (Figure 2 (b)).

Our main contributions are as follows.

- We propose a weakly supervised instance segmentation framework using two types of weak labels obtained without any additional manual annotation costs. Our method utilizes two types of weak labels to complement the lack of intercellular boundary information.
- We propose a single instance pasting to generate an intercellular boundary from two types of weak labels. A single instance is identified by cell counting and the intercellular boundary is then generated by pasting the detected single instance into another image.
- We evaluate our method under three conditions on a public dataset and demonstrate its state-of-the-art performance compared to conventional weakly supervised methods.

2. Related work

Cell segmentation: Traditionally, image processing-based methods using thresholding, level-set, and watershed have been utilized for automated cell segmentation [7, 37, 36, 4]. These methods need to be customized for each recognition target.

Deep learning-based cell segmentation methods have outperformed these image processing-based methods thanks to training with a certain amount of data [31, 6, 27, 33, 10, 24]. Ronneberger *et al.* proposed Unet, a fully convolutional network with skip-connection [31], and showed that it outperformed other image processing-based approaches in a cell tracking challenge dataset. However, these methods require annotations for each imaging condition, such as type of cells, type of microscopy, cultured condition, and density. The imaging conditions vary depending on the research field, so creating annotations for each condition is both time-consuming and labor-intensive.

Weakly supervised semantic segmentation: Weakly supervised semantic segmentation is the task that estimates class labels for each pixel (not, *i.e.,* distinguishing the same class instance). Most methods use an image-level annotation leverage class activation map [44] to generate pseudo segmentation labels [2, 17, 9]. The main focus of these methods is how to extend the CAM clues. For example, Ahn *et al.* [2] expand CAM clues by training an affinity net that learns inter-pixel semantic affinity from the clues. Thanks to recent developments, the boundaries of different class objects can be retrieved accurately using classification labels. However, the boundary of the same class object

is difficult to obtain since the class label does not contain much instance information.

Class labels and saliency detection labels have been used to improve this task [22, 39]. Lee *et al.* [22] have proposed a training strategy to extract segmentation information from these two labels. Xu *et al.* [39] improved the segmentation performance by jointly learning the affinity of the segmentation task and saliency detection task. However, there is no saliency detection label in the cell image.

Weakly supervised instance segmentation: Weakly supervised instance segmentation is the task that estimates the segment for each instance (*i.e.,* identifies the same class instance as a different instance). Various methods have used the class label [45, 1, 12], similar to semantic segmentation. For example, Zhou *et al.* [45] have proposed a peak response map to obtain instance clues from the class label. However, since these methods are designed for general images, they do not consider densely distributed objects, which makes it difficult to recognize cell images, as they often include such objects.

The bounding box label has sometimes been used [15, 38, 20, 34, 16]. The basic idea is to treat the weakly supervised instance segmentation task as a multi-instance learning problem and extend the object detection model (e.g., RCNN) to the instance segmentation model by multi-instance learning loss. However, these labels require much higher annotation costs than point-level and image-level annotation.

Weakly supervised segmentation for cell image: The bounding box and scribble labels are sometimes used for cell or nuclei segmentation tasks [8, 41, 21]. Lee *et al.* [21] proposed a method using scribble labels that selects pseudo-labels by the iteration average of the estimation results. Dong *et al.* [8] used bounding boxes and recognized object by using a peak response propagation.

Point-level annotation, which can be easily obtained rather than the bounding box, is widely utilized for weakly supervised cell or nuclei segmentation.

Some methods [29, 5, 30] have used color information by implementing color clustering to generate pseudo-labels. However, as these methods are designed for H&E stained images, which have high contrast, they do not work for phase-contrast images, which have low contrast.

Some methods utilize the learning ability of a neural network [42, 26]. Yoo *et al.* [42] used a shallow network to obtain edge information based on the assumption that the shallow network tends to extract edge information. Nishimura *et al.* [26] utilized a relevance map (*i.e.,* the relevant pixel for output) of a detection network. However, as these methods rely on the implicitly learned features of the network, they are Therefore, we could not adjust how well the neural network extracts boundary information.

In contrast to the weakly supervised methods discussed above, our method enables us to create the boundaries between cells ourselves and learn the boundaries directly.

Copy-and-paste augmentation: Copy-and-paste augmentation has been used as an effective data augmentation method for instance segmentation. The basic strategy is to copy some instances from one image and paste them onto another image [13, 11]. In contrast to methods that use copy-and-paste for supervised learning, our method utilizes it for weakly supervised learning, which enables us to complement the intercellular boundary information.

3. Weakly supervised cell segmentation

Overview: First, we train a foreground estimation network f_f using class labels and a cell detection network f_d using cell position labels. Next, we generate self-generated labels that include the intercellular boundary information from the foreground estimations and cell detection results. To create self-generated labels, we propose a single instance pasting that identifies single instances and pastes them to generate an intercellular boundary. We then train a boundary estimation network f_b with the self-generated labels. Instance segmentation is performed by combining the estimation results of the boundary estimation network f_b and the cell detection network f_d.

3.1. Networks training with two weak labels

In this section, we explain how to train a foreground estimation network f_f and a cell detection network f_d by utilizing weak labels and existing methods. We leverage CAM-based techniques [17] to train the foreground estimation network f_f. For the cell detection network f_d, we use a heatmap-based detection method [26].

Foreground estimation with image-level annotation: In this step, we train the foreground estimation network f_f from the class label. We first extract a foreground clue $\mathbb{C}_{fg}$ and background clue $\mathbb{C}_{bg}$ from a foreground activation map. $\mathbb{C}_{fg}$ and $\mathbb{C}_{bg}$ are sets of pixels estimated to be foreground or background, respectively. Then, we train the network f_f with the clues $\mathbb{C}_{fg}$ and $\mathbb{C}_{bg}$.

We follow the approach of Jo *et al.* [17] to obtain the foreground activation map. First, a classification network is trained with binary cross-entropy between a class output of the network and class label, the same as a normal classification problem. Given the input image $\mathbf{x}_i$, a feature map $\mathbf{M}_i \in \mathbb{R}^{\frac{W}{4} \times \frac{H}{4} \times K}$ is extracted by the ResNet-based feature extractor. Then, the class output is obtained from the feature map $\mathbf{M}_i$ by a global averaging operation. W and H are the width and height of the input image, and K is the number of classes. The feature map implicitly learns the foreground clues by the classification loss (binary cross-entropy and reconstruction loss) with class labels. In contrast to the method of Jo [17], where the aim is to extract an activation map for each class, we obtain a foreground activation map

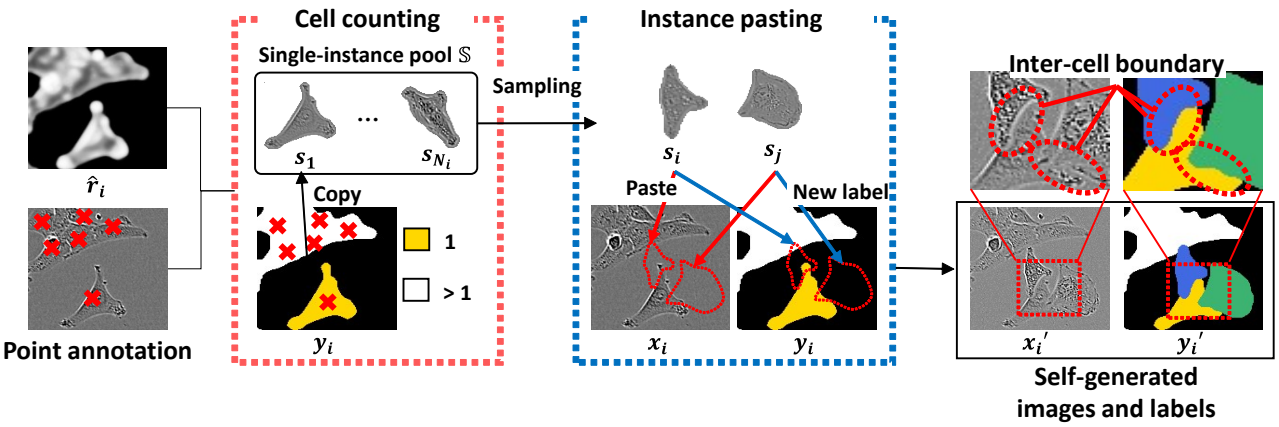

Figure 3. Overview of single instance pasting. Single cell regions (yellow $\mathbf{y}_i$ pixel) and multiple cell regions (white $\mathbf{y}_i$ pixel) are obtained by cell counting. The single instance is added to a single-instance pool $\mathcal{S}$, and then self-generated labels $\mathbf{x}_i'$ and $\mathbf{y}_i'$ are created by pasting the single-cell instances that are sampled from the single instance pool.

$\mathbf{M}_i^{fg} \in \mathbb{R}^{W \times H}$ by the max operation of class direction and a resize operation. Finally, we obtain the foreground clue $\mathbb{C}_{fg}$ and the background clue $\mathbb{C}_{bg}$ by foreground thresholding th_{fg} and background thresholding th_{bg} from $\mathbf{M}_i^{fg}$.

We train the foreground segmentation network f_f with the foreground clues $\mathbb{C}_{fg}$ and the background clues $\mathbb{C}_{bg}$. The loss is calculated only on the pixels in the clues, and the other pixels are ignored. The loss function of network f_f is defined as

$$L_{bseg} = \frac{1}{N_{fg}} \sum_{p \in \mathbb{C}_{fg}} -\log \hat{\mathbf{r}}_i(p) + \frac{1}{N_{bg}} \sum_{p \in \mathbb{C}_{bg}} -\log\left(1 - \hat{\mathbf{r}}_i(p)\right), \quad (1)$$

where $\hat{\mathbf{r}}_i = f_f(\mathbf{x}_i)$, p indicates the coordinate, and N_{fg} and N_{bg} are the number of foreground and background pixels. The network f_f is trained to output $\hat{\mathbf{r}}(p) = 1$ in the pixel in $\mathbb{C}_{fg}$, and $\hat{\mathbf{r}}(p) = 0$ in the pixel in $\mathbb{C}_{bg}$ by this loss function. If a pixel does not belong to both sets , it is not used for the loss calculation.

Cell detection with nuclei positions: To obtain cell positions, we use a heatmap-based cell detector [26] that can be trained with cell position labels. The cell detection network f_d is trained to output the cell position heatmap $\mathbf{h}_i$, and then cell positions are obtained by taking the peak of the estimated heatmap $\hat{\mathbf{h}}_i$. An example of the estimated heatmap is shown in Figure 4 $\hat{\mathbf{h}}_i$. The heatmap $\mathbf{h}_i$ is generated by applying a 2D Gaussian filter on the annotated cell positions. Then, f_d is trained with the MSE loss between the heatmap $\mathbf{h}_i$ and an estimated heatmap $\hat{\mathbf{h}}_i$, as $L_{det} = MSE(\mathbf{h}_i, \hat{\mathbf{h}}_i)$. After training, the cell positions p_i can be obtained by taking the local maximum of the estimation $\hat{\mathbf{h}}_i$.

3.2. Label generation with single instance pasting

The foreground and cell positions are obtained by the above process, but the results do not contain intercellular boundary information. We therefore propose a single instance pasting to train a boundary estimation network f_b. We design the single instance pasting so that the self-generated label contains two types of information. The first is the intercellular boundary information. Since the weak labels do not contain intercellular boundary information, we generate the boundary by pasting. The second is unknown boundary region information (multiple cell regions). Although multiple cell regions include intercellular boundary information, we do not know the boundary. We identify these regions so that they can be ignored.

Figure 3 shows an overview of the single instance pasting, which consists of two steps. First, we find single-cell regions and multiple-cell regions by cell counting and add the single-cell regions to a single-instance pool. Second, we generate intercellular boundaries by pasting the instances that are sampled from the pool.

Cell counting: Given the estimated foreground region $\hat{\mathbf{r}}_i$ and the cell positions p_i, we count cells in the foreground segment (*i.e.,* the connected component of $\hat{\mathbf{r}}_i$) to identify the single-cell regions and multiple-cell regions. The single-cell regions are used for intercellular boundary generation, and the multiple-cell regions are used to ignore the loss calculation of unknown regions in the final training step.

The counting provides the single-cell regions (yellow $\mathbf{y}_i$ pixels in Figure 3) and the multiple-cell regions (white pixels). We add the pixels in the multiple-cell regions to a set of ignoring pixels $\mathbb{I}$ (white $\mathbf{y}_i$ pixels in Figure 3). We generate an initial label $\mathbf{y}_i$ by giving a label to the single-cell region, as indicated by the yellow $\mathbf{y}_i$ pixel in Figure 3. We then add an instance image $\mathbf{s}_k$ into a single instance pool $\mathcal{S}$, where $\mathbf{s}_k$ is generated by masking the input image $\mathbf{x}_i$ with the single instance region (*e.g.,* $\mathbf{s}_k$ in Figure 3). We omit the instance when the instance size is too small or too large,

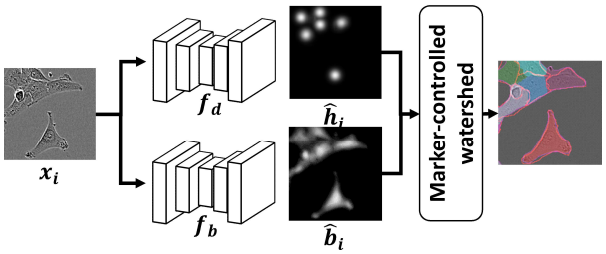

Figure 4. Instance segmentation process. First, the heatmap $\hat{\mathbf{h}}_i$ and boundary map $\hat{\mathbf{b}}_i$ are obtained from input image $\mathbf{x}_i$. Then, an instance segmentation result is obtained by using marker-controlled watershed.

specifically if the width or height of the instance is smaller than th_{sm} or larger than th_{la}.

Instance pasting: Given the input image $\mathbf{x}_i$, the initial label $\mathbf{y}_i$, and the set of N_c sampled instances $\{\mathbf{s}_i, ..., \mathbf{s}_j\}$, our single-instance pasting generates the pasted image x'_i and the updated instance label $\mathbf{y}'_i$. N_c is the number of samples . The cell is captured as a series of time-lapse images, where the appearance of the cell differs between the initial state and the late state. We prepare the image, label, and paste instances from a close frame to generate a natural image. The input image $\mathbf{x}_i$ and the initial label $\mathbf{y}_i$, which are generated in cell counting, are selected from the image that contains a certain amount of single-cell regions (*i.e.,* a pair containing more than th_{ss} single instance pixels). To sample the pasting instance from a close frame to the image and label pair, we divide the instance pool $\mathcal{S}$ into several subset pools by time. Specifically, we generate the subset pool every th_t hours, and we sample the N_c instances $\mathbf{s}_i$ from the subset pool to which the selected frames belong. We then generate the pasted image $\mathbf{x}'_i$ by randomly pasting instances $\{\mathbf{s}_i, ..., \mathbf{s}_j\}$ into the input image $\mathbf{x}_i$, as indicated by the red arrow in Figure. 3. The updated instance label $\mathbf{y}'_i$ is generated by giving a new label to the pasted regions of the initial label $\mathbf{y}_i$ (blue arrow). The red dotted circle shows an example of the generated boundary.

3.3. Instance segmentation

To achieve the instance segmentation, we first train a boundary estimation network f_b by the self-generated images and labels. Then, we perform instance segmentation by combining the boundary estimation result $\hat{\mathbf{b}}_i$ and the detection result $\hat{\mathbf{h}}_i$.

Training with self-generated label: We use distance formed representation [3] for the boundary estimation network f_b. An example of a distance formed boundary map is shown in Figure 4 $\hat{\mathbf{b}}_i$, where the center of the cell has a high pixel value that gradually decreases towards the boundaries.

The ground-truth of distance formed boundary map $\mathbf{b}_i$ is generated by taking the max of the normalized distance map of each instance [3]. We do not calculate the loss on multiple cell regions since we do not know their accurate boundaries. We expect the network to implicitly learn the boundaries from self-generated images and labels by ignoring the unknown regions. The approach is inspired by the segmentation training using CAM [17, 2], which trains a segmentation model with high-confidence foreground clues and background clues (like Eq. 1).

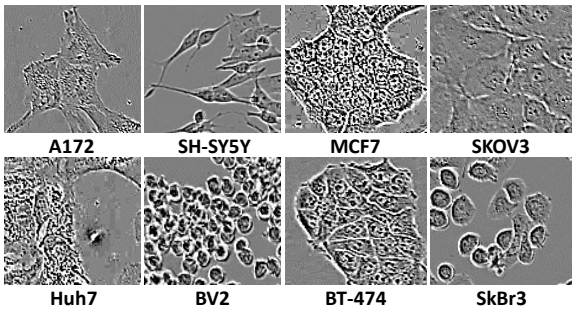

Figure 5. Examples of images in LIVECell dataset [10]. The appearance of cells varies depending on the cell type, including spherical morphology (*e.g.,* BV2 and SkBr3) and adherent morphology (*e.g.,* A172, SKOV3, and Huh7).

The loss function of the boundary estimation network f_b is defined as

$$L_b = \frac{1}{N_{nm}} \sum_{p \notin \mathbb{I}} (\mathbf{b}(p) - \hat{\mathbf{b}}(p))^2, \tag{2}$$

where $\hat{\mathbf{b}}_i$ is the estimation result of f_b, $\mathbb{I}$ is the set of coordinates in multiple cell regions, and N_{nm} is the number of pixels that are not on multiple regions. As indicated by $\hat{\mathbf{b}}_i$ in Figure 4, the pixel value on the intercellular boundary reaches a low value after the training.

Instance segmentation: We perform instance segmentation by using the boundary estimation network f_b and the cell detection network f_d. As shown in Figure 4, given the input image $\mathbf{x}_i$, the heatmap $\hat{\mathbf{h}}_i$ and distance formed boundary map $\hat{\mathbf{b}}_i$ are estimated by f_d and f_b, respectively. Then, we combine these estimations by marker-controlled watershed [25]. We treat the heatmap as a marker and the distance estimation as an input image.

4. Experiments

Implementation details: We implemented our method using the PyTorch framework [28]. We used ResNet 50 [14] pretrained with ImageNet [19] for the classification network (used for the CAM extraction). We utilized the Unet [31] architecture for the cell detection network f_d, the foreground estimation network f_f, and the boundary estimation network f_b. The foreground threshold th_{fg} and background threshold th_{bg} were 0.3 and 0.2, respectively. We set the number of pasting instances to $N_c = 3$, the minimum size of cell $th_{sm} = 10$, the maximum size of cell $th_{la} = 200$,

Method	L	A172	BT474	BV2	Huh7	MCF7	SHSY5Y	SkBr3	SKOV3	Avg.
Chalfoun [4]	U	0.892	0.809	0.672	**0.838**	**0.891**	0.766	0.868	0.823	0.820
Qu [29]	P	0.139	0.242	0.656	0.103	0.335	0.086	0.576	0.163	0.288
Nishimura [26]	P	0.790	**0.832**	**0.816**	0.458	0.865	0.753	**0.873**	0.725	0.764
Ours	P, I	**0.921**	0.814	0.809	0.775	0.880	**0.788**	0.842	**0.860**	**0.836**
Ronneverger [31]	F	0.844	0.835	0.766	0.880	0.743	0.541	0.897	0.879	0.798
Edlund [10]	F	0.938	0.890	0.886	0.914	0.905	0.844	0.942	0.940	0.907

Table 1. Quantitative evaluation results of instance segmentation on $F1_p$ for each cell type. Avg. is the average performance of whole-cell types. L indicates label conditions: U is unsupervised, I is image-level annotation, P is point-level annotation, and F is fully supervised. The boldface indicates the best performance in the weakly supervised setting.

Method	L	A172	BT474	BV2	Huh7	MCF7	SHSY5Y	SkBr3	SKOV3	Avg.
Chalfoun [4]	U	0.570	0.563	0.483	0.494	0.561	**0.479**	**0.766**	0.502	0.552
Qu [29]	P	0.198	0.321	**0.659**	0.203	0.385	0.106	0.589	0.198	0.333
Nishimura [26]	P	0.624	**0.690**	0.541	0.513	0.502	0.426	0.612	0.640	0.568
Ours	P, I	**0.678**	0.565	0.643	**0.608**	**0.577**	0.449	0.649	**0.773**	**0.618**
Ronneverger [31]	F	0.761	0.747	0.767	0.789	0.709	0.544	0.861	0.821	0.750
Edlund [10]	F	0.779	0.788	0.655	0.830	0.644	0.623	0.806	0.856	0.748

Table 2. Quantitative evaluation results of instance segmentation on $Dice_o$ for each cell type. Avg. is the average performance of whole-cell types. L indicates label conditions: U is unsupervised, I is image-level annotation, P is point-level annotation, and F is fully supervised. The boldface indicates the best performance in the weakly supervised setting.

the single cell region threshold $th_{ss} = 500$, and the hours of the subset pool $th_t = 12$.

Random crop and rotation were used for the data augmentation. We used the Adam [18] optimizer with the learning rate = 1e-3 and the mini-batch size of 16 for all networks. The classification network and detection network f_d were trained by early stopping based on the classification loss and the detection loss of validation, respectively. The foreground estimation network f_f was trained with 30 epochs. The boundary estimation network f_b was trained by early stopping based on the loss of self-generated labels of validation.

Dataset: We used the LIVECell dataset [10] to evaluate our method. This dataset contains eight types of cells captured by phase-contrast microscopy with 520 $\times$ 704 resolution. The cells were cultured from early seeding to full confluence. Unlike other cell segmentation datasets such as the cell tracking challenge [35] and BBBC datasets [23], LIVECell has variations in cell type and density. As shown in Figure 5, the cells have various appearances depending on cell type. The bounding box and the instance mask were manually annotated for each image. The total number of training, validation, and test data were 3188, 569, 1548. To train our method, we treated cell types as the class label and the center of the bounding box as a cell position label.

Metrics: We used pixel-level F1 score $F1_p$ and object-level Dice coefficient $Dice_o$ [32] to evaluate the performance of binary segmentation and instance segmentation, respectively. $F1_p$ is calculated by $F1_p = \frac{2TP}{(2TP+FP+FN)}$, where TP, FP, and FN are the number of true positives, false positives, and false negatives (determined by the foreground and background labels). $Dice_o$ is defined as

$$Dice_o = \frac{1}{2}(\sum_{i=1}^{N_g} \gamma_i Di(g_i, p_{g_i}) + \sum_{j=1}^{N_p} \gamma_j Di(p_j, g_{p_j}), \quad (3)$$

where g_i is the ith ground-truth object, p_j is the jth predicted object, p_{g_i} and g_{p_j} are the matched object of the predicted object and ground-truth object, Di is a dice operation, and N_g and N_p are the number of ground-truth objects and predicted objects, $\gamma_i = \frac{|g_i|}{\sum_{i=1}^{N_g}|g_i|}$ and $\gamma_j = \frac{|p_j|}{\sum_{j=1}^{N_p}|p_j|}$, respectively. The object-level dice is calculated as the dice for each object by weighting it according to the size of the object by γ_i and γ_j.

4.1. Comparisons

We compared our method with the following five conventional methods. 1) **Chalfoun** *et al.* [4]: Image processing-based instance segmentation method, which uses the gradient of the image for segmentation (unsupervised). 2) **Qu** *et al.* [29]: Weakly supervised nuclei segmentation method, which uses Voronoi diagram and color clustering with point-level annotation to calculate loss (weakly supervised). 3) **Nishimura** *et al.* [26]: Weakly supervised cell instance segmentation method, which uses the relevant pixels for the detection of the segmentation (weakly supervised). 4) **Ronneberger** *et al.* [31]: Well-known supervised segmentation method (Unet), which trains the network by using weighted cross-entropy (supervised). 5) **Edlund** *et al.* [10]: R-CNN-based instance segmentation method. The weakly supervised methods are trained with the nuclei po-

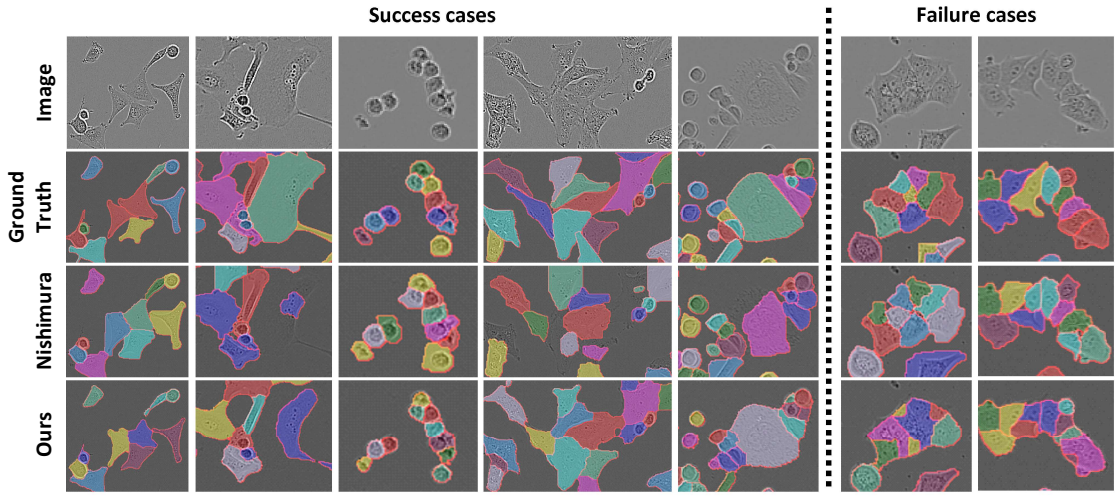

Figure 6. Example of estimation results. The color indicates the instance.

sitions, and the supervised methods are trained with the labeled data that is annotated for each cell boundary. We used the same number of training and validation data for the training.

Tables 1 and 2 list the performance of each method in terms of $F1_p$ and $Dice_o$. The method of Qu *et al.* [29] relies on the color of images, and so it cannot capture an accurate boundary of the cell. Since it is designed for H&E stained images, the method cannot work on a phase-contrast image. The method of Nishimura *et al.* [26], which uses the relevance pixel of the detection network, outperforms Chalfoun [4] in terms of $Dice_o$. Our method outperforms these weakly supervised methods on both metrics on average. Compared with the supervised methods, our method is inferior on $Dice_o$. In terms of $F1_p$, our method outperforms the method of Ronneberger *et al.* [31]. Their method uses weighted cross-entropy, which gives high weights to the bounding pixels rather than other foreground pixels. As a result, the pixels around the boundary tend to be recognized as the background, which decreases the $F1_p$.

Figure 6 shows the qualitative results of Nishimura's method and ours, where the five columns on the left show success cases and the two on the right show failure cases. Since both methods use a detection network, there is no difference in the detection results. Regarding the accuracy of the boundaries in the success cases, our method outperforms Nishimura's. Their method uses relevant pixels that contribute to detecting cells, and since the cell boundary does not always contribute to the detection, the boundaries are sometimes over- or under-estimated. In contrast, our method tries to directly learn the boundary by the self-generated labels, and as a result, the boundary estimation is accurate.

The failure cases in Figure 6 reveal the limitation of the proposed method. In some cell types (e.g., BT-474 and MCF7), the cell morphology changes upon contact with other cells. The cells of the failure cases have this cell morphology, which is different from the morphology of a single cell. Therefore, the single instance pasting cannot generate a similar label, and our method is not able to deal with this type of image.

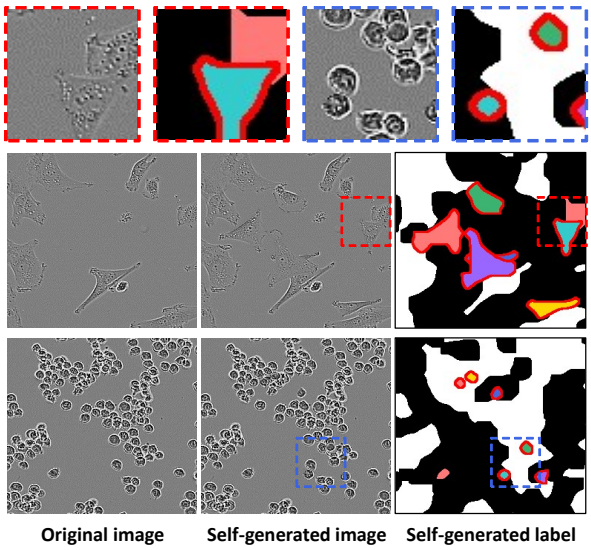

Figure 7. Example of self-generated images and labels. White pixel indicates multiple-cell region and colors indicate single instance label. Red outline indicates pasted instance. Top images are enlarged images of red and blue dotted rectangles.

4.2. Ablation study

To check the image quality generated by single instance pasting, we show examples of the self-generated images

Method	N_c	$F1_p$	$Dice_o$
w/o sip	–	0.738	0.532
Ours	1	0.814	0.579
	3	**0.836**	**0.618**
	5	0.832	**0.618**
	7	0.830	0.613

Table 3. Ablation study.

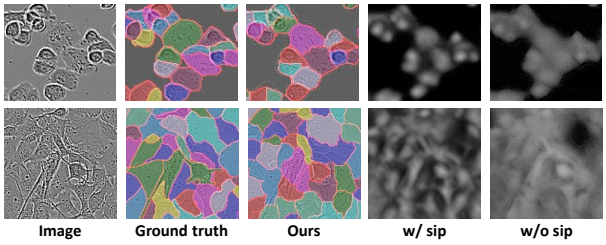

Figure 8. Example of boundary estimation outputs.

and their labels in Figure 7. The white pixels of a self-generated label indicate multiple cell regions and the color means each instance label. The red outline indicates the pasted instance label. As we can see, the appearance of the self-generated images in Fig. 7 looks natural. Unlike general images that capture different light or scale conditions, the phase-contrast image is captured under the same condition, which makes the appearance of the pasted image more natural.

To investigate the effectiveness of our single instance pasting, we tested it without single instance pasting (w/o sip in Table 3). In this setting, the boundary estimation network f_b is trained with the initial label y_i while ignoring the multiple-cell region. In addition, we examined the relation between the number of pasting instances N_c and the performance. Table 3 shows the average performance of each cell type. We can see here that the performance in both metrics was increased by the single instance pasting. $N_i = 3$ is the best performance among other settings. By increasing the number of pasted instances, the boundary estimation network can better learn the intercellular boundaries. The single instance pasting works robustly on $N_i > 3$.

Figure 8 shows examples of the distance-formed boundary map of network $\hat{d}$ on the test data. The two columns on the right show estimation results with and without the single instance pasting. In the case of single instance pasting, the output captures rough cell shapes, even if they are dense. In contrast, the output of w/o sip could not estimate the cell boundary under the dense condition. Without the single instance pasting, the intercellular boundaries cannot be learned and therefore the method does not work under the dense condition.

Method	$F1_p$	$Dice_o$
Chalfoun [26]	0.892	0.570
Qu [29]	0.246	0.271
Nishimura [26]	0.481	0.435
Ours	**0.920**	**0.650**

Table 4. Comparison of weak or unsupervised methods on application setting.

4.3. Application

As mentioned in the introduction, our method enables the class label and point label to be obtained without any additional manual annotations. To demonstrate the effectiveness of the proposed method in a realistic situation, we trained it with labels that were obtained in this manner.

The LIVECell dataset [10] includes paired phase-contrast and fluorescent images that can be obtained by capturing the cells stained with nuclei. The point labels (i.e., cell position) were obtained from the fluorescent images by using thresholding and finding local maxima. Since the type of cells used was recorded, the class labels can be obtained without additional annotation cost. Cell types A172 and A549 were used to capture the paired images. Therefore, there are two class labels. The dataset includes 798 paired images with 1408 $\times$ 1040 resolution. We used 157 manually annotated images for the test data. The images includes A172 cells and are the same as the images used for the comparisons in Section 4.1. We compared our method with the same weakly supervised methods that were used in Section 4.1.

Table 4 shows the performance comparisons. Compared to the results discussed in Section 4.1, the performance of Nishimura's method [26] decreased dramatically. Their method relies on the relevance map of the detection network, so when there are fewer cell types in the training data, the cell shape is not used for detection and the cell shape cannot be estimated. In contrast, the performance of our method did not decrease, which demonstrates its effectiveness for realistic use cases.

5. Conclusion

In this paper, we proposed a weakly supervised cell segmentation method with two types of weak labels obtained without additional manual annotation costs. We generated intercellular boundaries ourselves by pasting a single cell to the original image to obtain the intercellular boundary label from two weak labels. Experiments on a public dataset demonstrated that our method achieves a state-of-the-art performance compared to conventional weakly supervised methods.

Acknowledgements: This work was supported by JSPS KAKENHI Grant Number JP21J21810, JP20H04211, and JST ACT-X Grant Number JPMJAX21AK, Japan.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, pages 2209–2218, 2019.

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, pages 4981–4990, 2018.

[3] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, pages 5221–5229, 2017.

[4] Joe Chalfoun, Michael Majurski, Alden Dima, Christina Stuelten, Adele Peskin, and Mary Brady. Fogbank: a single cell segmentation across multiple cell lines and image modalities. *Bmc Bioinformatics*, 15(1):1–12, 2014.

[5] Alireza Chamanzar and Yao Nie. Weakly supervised multi-task learning for cell detection and segmentation. In *ISBI*, pages 513–516, 2020.

[6] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. Dcan: deep contour-aware networks for accurate gland segmentation. In *CVPR*, pages 2487–2496, 2016.

[7] Eric Cosatto, Matt Miller, Hans Peter Graf, and John S Meyer. Grading nuclear pleomorphism on histological micrographs. In *ICPR*, pages 1–4, 2008.

[8] Meng Dong, Dong Liu, Zhiwei Xiong, Xuejin Chen, Yueyi Zhang, Zheng-Jun Zha, Guoqiang Bi, and Feng Wu. Instance segmentation from volumetric biomedical images without voxel-wise labeling. In *MICCAI*, pages 83–91, 2019.

[9] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*, pages 4320–4329, 2022.

[10] Christoffer Edlund, Timothy R Jackson, Nabeel Khalid, Nicola Bevan, Timothy Dale, Andreas Dengel, Sheraz Ahmed, Johan Trygg, and Rickard Sjögren. Livecell—a large-scale dataset for label-free live cell segmentation. *Nature methods*, 18(9):1038–1045, 2021.

[11] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, pages 682–691, 2019.

[12] Weifeng Ge, Sheng Guo, Weilin Huang, and Matthew R Scott. Label-penet: Sequential label propagation and enhancement networks for weakly supervised instance segmentation. In *ICCV*, pages 3345–3354, 2019.

[13] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, pages 2918–2928, 2021.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. *NeurIPS*, 32, 2019.

[16] Jaedong Hwang, Seohyun Kim, Jeany Son, and Bohyung Han. Weakly supervised instance segmentation by deep community learning. In *WACV*, pages 1020–1029, 2021.

[17] Sanghyun Jo and In-Jae Yu. Puzzle-cam: Improved localization via matching partial and full features. In *ICIP*, pages 639–643, 2021.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012.

[20] Shiyi Lan, Zhiding Yu, Christopher Choy, Subhashree Radhakrishnan, Guilin Liu, Yuke Zhu, Larry S Davis, and Anima Anandkumar. Discobox: Weakly supervised instance segmentation and semantic correspondence from box supervision. In *ICCV*, pages 3406–3416, 2021.

[21] Hyeonsoo Lee and Won-Ki Jeong. Scribble2label: Scribble-supervised cell segmentation via self-generating pseudo-labels with consistency. In *MICCAI*, pages 14–23, 2020.

[22] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*, pages 5495–5505, 2021.

[23] Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637–637, 2012.

[24] Filip Lux and Petr Matula. Cell segmentation by combining marker-controlled watershed and deep learning. *arXiv:2004.01607*, 2020.

[25] Peer Neubert and Peter Protzel. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In *ICPR*, pages 996–1001, 2014.

[26] Kazuya Nishimura, Chenyang Wang, Kazuhide Watanabe, Ryoma Bise, et al. Weakly supervised cell instance segmentation under various conditions. *Medical Image Analysis*, 73:102182, 2021.

[27] Hirohisa Oda, Holger R Roth, Kosuke Chiba, Jure Sokolić, Takayuki Kitasaka, Masahiro Oda, Akinari Hinoki, Hiroo Uchida, Julia A Schnabel, and Kensaku Mori. Besnet: boundary-enhanced segmentation of cells in histopathological images. In *MICCAI*, pages 228–236, 2018.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

[29] Hui Qu, Pengxiang Wu, Qiaoying Huang, Jingru Yi, Zhennan Yan, Kang Li, Gregory M Riedlinger, Subhajyoti De, Shaoting Zhang, and Dimitris N Metaxas. Weakly supervised deep nuclei segmentation using partial points annotation in histopathology images. *IEEE transactions on medical imaging*, 39(11):3655–3666, 2020.

[30] Hui Qu, Jingru Yi, Qiaoying Huang, Pengxiang Wu, and Dimitris Metaxas. Nuclei segmentation using mixed points and masks selected from uncertainty. In *ISBI*, pages 973–976, 2020.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[32] Korsuk Sirinukunwattana, David RJ Snead, and Nasir M Rajpoot. A stochastic polygons model for glandular structures in colon histology images. *IEEE transactions on medical imaging*, 34(11):2366–2378, 2015.

[33] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[34] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *CVPR*, pages 5443–5452, 2021.

[35] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature methods*, 14(12):1141–1152, 2017.

[36] Mitko Veta, Paul J Van Diest, Robert Kornegoor, André Huisman, Max A Viergever, and Josien PW Pluim. Automatic nuclei segmentation in h&e stained breast cancer histopathology images. *PloS one*, 8(7):e70221, 2013.

[37] Tomas Vicar, Jan Balvan, Josef Jaros, Florian Jug, Radim Kolar, Michal Masarik, and Jaromir Gumulec. Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison. *BMC bioinformatics*, 20(1):1–25, 2019.

[38] Juan Wang and Bin Xia. Bounding box tightness prior for weakly supervised image segmentation. In *MICCAI*, pages 526–536, 2021.

[39] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *ICCV*, pages 6984–6993, 2021.

[40] Guanyu Yang, Chuanxia Wang, Jian Yang, Yang Chen, Lijun Tang, Pengfei Shao, Jean-Louis Dillenseger, Huazhong Shu, and Limin Luo. Weakly-supervised convolutional neural networks of renal tumor segmentation in abdominal cta images. *BMC Medical Imaging*, 20(1):1–12, 2020.

[41] Lin Yang, Yizhe Zhang, Zhuo Zhao, Hao Zheng, Peixian Liang, Michael TC Ying, Anil T Ahuja, and Danny Z Chen. Boxnet: Deep learning based biomedical image segmentation using boxes only annotation. *arXiv preprint arXiv:1806.00593*, 2018.

[42] Inwan Yoo, Donggeun Yoo, and Kyunghyun Paeng. Pseudoedgenet: Nuclei segmentation only with point annotations. In *MICCAI*, pages 731–739, 2019.

[43] Tianyi Zhao and Zhaozheng Yin. Weakly supervised cell segmentation by point annotation. *IEEE Transactions on Medical Imaging*, 40(10):2736–2747, 2020.

[44] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[45] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, pages 3791–3800, 2018.