Transfer Learning-based Neuronal Cell Instance Segmentation with Pointwise Attentive Path Fusion

GEHUI LI\textsuperscript{1} and ZHIFENG XIAO\textsuperscript{2}

\textsuperscript{1}International School of Information Science & Engineering, Dalian University of Technology, Dalian, Liaoning 116620, China (G.L.)
\textsuperscript{2}School of Engineering, Penn State Erie, The Behrend College, Erie, PA 16563, USA (Z.X.)

Corresponding authors: Zhifeng Xiao (e-mail: zux2@psu.edu).

ABSTRACT  Accurate instance segmentation is essential for the behavior and morphology analysis of neuronal cells. The main challenges of this segmentation task involve irregular and concave cell morphology, low contrast on cell boundaries, cell clustering and adhesion, and the background noise in the phase contrast microscopy (PCM) images. To address these challenges, we propose a learning pipeline with three performance boosters that have not been extensively explored in prior studies, including transferring knowledge from a model pre-trained on a larger but similar dataset, enhancing the contrast of cells in a PCM image, and a pointwise attentive path fusion module that allows the learning model to capture informative features from critical areas. Experiments have been conducted on the Sartorius Cell Instance Segmentation dataset with three neuronal cell lines. Results show that the final model, with three boosters enabled, brings a mAP gain of 10.3%. Compared to the top three places in the leaderboard, our method shows comparable performance without using any ensemble method, making our model the state-of-the-art solution among the single model-based methods.

INDEX TERMS  datasets; neural networks; medical images process; transfer learning.

I. INTRODUCTION

Analysis of behavior and morphology for neuronal cells is an essential task to uncover mysteries in neural science \cite{1}. The underlying cellular mechanisms engaging neurons, astrocytes, and oligodendrocytes remain to be explored \cite{2}. From the lineage perspective, neuronal cells frequently sample the environment by interacting with the neighboring cells via their lamellipodia and filopodia, and these contacts are accompanied by a series of cellular dynamics such as movement, mitosis, and morphology changes. Thus, it is highly expected to have a tool that can accurately capture interactions between cells to better understand the neuronal cell behavior and normal brain development. Also, Neurological disorders \cite{3,4}, such as brain tumors, Parkinson’s and Alzheimer’s disease, have become a serious cause of disability and death across the world. To pursue effective treatments, it is crucial to understand and quantify how these disorders respond to medication.

One non-invasive and accessible approach is to employ light microscopy to analyze the behavior, population, and shape of neuronal cells \cite{5}. Microscopy imaging techniques, including differential interference contrast microscopy \cite{6} and phase contrast \cite{7}, can be adopted to capture cell appearance without staining, which drives the advancement of computer-aided morphology and behavior analysis for cells. With the assistance of real-time imaging techniques, such cell-level interactions can be captured and analyzed via common computer vision (CV) tasks, such as detection, segmentation, and tracking, among which accurate segmentation is crucial to pinpoint the cell contact time.

Current instance segmentation solutions have limited precision for neuronal cells. In prior study, a large cell segmentation dataset, named LIVECell \cite{8}, was developed. LIVECell consists of over 5K phase contrast microscopy (PCM) cell images with over 1.6 million individual cell annotations and covers eight cell lines. A comprehensive evaluation on the instance segmentation models shows that the neuroblastoma cell line Shsy5y consistently presents the lowest precision compared to the other seven tested cell types. A potential reason is that neuronal cells have an unusual, irregular and concave morphology, making them challenging to segment with commonly used segmentation models.
In addition to morphology, cell instance segmentation has other challenges such as low contrast on cell boundaries, cell clustering and adhesion, and background noise appearing in the PCM images. Thus, existing instance segmentation algorithms generally do not perform well on cell instance segmentation. Recent advances have witnessed the rise and wide application of deep learning models, which have gained worldwide success across almost all industries [9]. Deep neural network (DNN)-based models have been applied to tackle a wide spectrum of CV tasks such as image classification [10], object detection/tracking [11], [12], semantic/instance segmentation [13]. DNN-based instance segmentation algorithms have been extensively applied to cell instance segmentation. However, existing studies have not thoroughly explored the characteristics of neuronal cells with effective custom solutions. To fill this gap, we propose a DNN-based learning pipeline that focuses on instance segmentation for neuronal cells. The proposed learning system consists of three performance boosters, including knowledge transfer from a segmentation model pre-trained on the LIVECell dataset, applying an image contrast enhancement algorithm to adjust the contrast of PCM images, and a pointwise attentive path fusion (PAPF) module that allows the backbone network to identify and focus on the critical areas in the feature map that carry more informative patterns. The joint effect of the three boosters have brought significant and consistent performance gains.

The main contributions are summarized as follows.

- We propose a learning pipeline for neuronal cell instance segmentation with three performance boosters. First, we pre-train a Mask RCNN model on the LIVECell dataset, which shares a certain degree of similarity with the target dataset, namely, the Sartorius Cell Instance Segmentation (SCIS) dataset. Second, we adopt an image contrast enhancement algorithm to pre-process the raw PCM images so that the low contrast of cell boundaries and bodies can be improved. Lastly, we develop the PAPF module, which is a point-wise attention mechanism that can be easily integrated into the backbone network, i.e., the path aggregation network (PAN) in our case.

- The proposed learning pipeline has been extensively evaluated on the SCIS dataset, which focuses on neuronal cell instance segmentation. Results show that our final model can outperform the base model by 10.3%. Also, compared to the top three places in the leaderboard, our method shows comparable performance without using any ensemble method, making our model the state-of-the-art solution among the single model-based methods.

The rest of this paper is organized as follows. Section II reviews the related work and highlights the novelty of this study. Section III covers the dataset and a detailed and module-by-module description of the proposed method. Section IV describes the experimental design and reports the key results. Lastly, Section V summarizes the paper and point out future research directions.

II. RELATED WORK

A. DNN-BASED IMAGE SEGMENTATION

A wide spectrum of DNN-based models have been proposed in recent years. These models can be classified into the following categories based on their model architectures.

- Fully Convolutional Network (FCN) [14] is a milestone of DNN-based model for semantic segmentation. An FCN consists of pure convolutional layers, allowing the network to produce a segmentation map with the same size as the input image. FCN shows that DNNs can be trained to solve the segmentation task in an end-to-end manner. However, a drawback of FCN is its high computational cost, limiting its usage for real-time systems.

- Another line of models utilize an encoder-decoder architecture that generally consists of two parts, namely, an encoder and a decoder. Representative models include SegNet [15], HRNet [16], U-Net [17], Linknet [18] and V-Net [19]. These models differ in the architectural design of the encoder and decoder. Multiple design strategies have been adopted, such as deconvolution [18], connection of high-to-low resolution convolution streams [16], and contracting & expanding paths [17].

- Multiscale and Pyramid Network based models were developed to capture cross-scale features. A representative model in this category is the Feature Pyramid Network (FPN) [20], which consists of a bottom-up and a top-down pathway with lateral connections to fuse low and high resolution features. Extensions of FPN include Pyramid Scene Parsing Network [21], Adaptive Pyramid Context Network [22], and salient object segmentation [23].

- RCNN based models refer to a family of Regional CNN models that have been successful in object detection. Faster RCNN [24] adopts a region proposal network (RPN) to generate a collection of object candidates, called the Region of Interests (RoIs), which are further normalized by an RoIPool layer before sent to the detection head. Mask RCNN [25] modifies Faster RCNN by adding a segmentation branch to predict the mask. The Path Aggregation Network (PANet) [26] extends Mask RCNN by adding an augmented bottom-up pathway to the FPN backbone to better capture features from lower layers.

- Attention-based models have recently become popular. Chen et al. [27] adopt an attention mechanism to learn weights for multiscale features at each pixel location. Huang et al. propose a Reverse Attention Network [28] that can be trained to identify the features not associated with a target class. Li et al. propose a Pyramid Attention Network [29] to exploit global contextual features.
B. CELL INSTANCE SEGMENTATION

We investigate several prior studies that employ DNN-based models for cell instance segmentation. Yi has been the first author for a series of work. In [30], a hierarchical neural network comprising object detection and segmentation modules was developed, aiming to make full use of features at multiple levels to capture the contours of neuronal cells, as well as the filopodia and lamellipodia. Another work of Yi [31] proposes a context-refined neuronal cell instance segmentation model to well differentiate adjacent cells with large parts in the same RoI. A follow-up work [32] investigates a box-based cell instance segmentation method that takes advantage of keypoints detection, which suggest the locations of bounding boxes as well as segmentation masks. In another work, Yi et al. [33] integrate an attention mechanism into a combined network of a single shot multi-box detector [34] and a U-Net. Prangemeier et al. develop attention-based cell detection transformer [35], which firstly applied transformer-based models to cell instance segmentation. Nishimura et al. propose a weakly supervised cell instance segmentation method [36] that allows the model to be trained with rough cell centroid positions as training data, which can greatly reduce the annotation cost.

Our investigation shows that the proposed method differs from the existing DNN-based cell instance segmentation methods in three aspects. First, the effect of transfer learning has not been extensively studied. The development of LIVECell brings a new opportunity to explore how well knowledge can be transferred from LIVECell to SCIS, given the similarity between the two domains. Second, low contrast on the boundaries and bodies of cells has been a challenge for effective segmentation. Prior studies have not pursued the direction of contrast enhancement. Thus, we adopt an existing contrast enhancement algorithm that can keep the well-exposed pixels unchanged and enhance the under-exposed ones. Lastly, we develop a custom attention module that enables multiscale and pointwise attention for path fusion, which has not been seen in the literature. These design ideas distinguish our work from existing studies and have been validated via extensive experiments.

III. MATERIAL AND METHODS

A. DATASET

In this study, we employ the Sartorius Cell Instance Segmentation (SCIS) dataset, which was from a recent Kaggle competition [1] ended on December 30, 2021. SCIS is the successor of the LIVECell dataset, which is a large, high-quality, manually annotated, and expert-validated dataset of phase-contrast images for cell instance segmentation. LIVECell contains eight cell types, as shown in Figure [1] and the number of each cell line is in the range of 600 and 735, with a total of 5,239 images and 1,686,353 individual cell annotations. It is observed that cells with different types present different characteristics in size, density, and shape.  

Among the eight cell types, Shsy5y, a thrice-subcloned cell line derived from the SK-N-SH neuroblastoma cell line, presented the lowest segmentation precision due to its unique neuronal morphology with long protrusions and overlapping cells. This segmentation challenge drives the development of SCIS, a dataset that focuses on instance segmentation for neuronal cells.

FIGURE 1: LIVECell dataset cell type counts (the number marked within each bar represent the count for each cell type). The included cell lines are Mcf7 (breast cancer), Shsy5y (Neuroblastoma), Skbr3 (ovarian cancer), Bt474 (breast cancer), A172 (glioblastoma), Bv2 (Microglia), Skov3 (breast cancer), Huh7 (hepatocellular), in which Bv2 species are from mouse, and the rest are from human.

The SCIS dataset consists of three neuronal cell types, including Cort, Shsy5y, and Astro. We report the statistical information of the three cell lines in Table 1. It is noted that the dataset consists of a total of 606 PCM image samples, including 320 Cortical neurons (Cort), 155 Shsy5y, and 131 astrocytes (Astro) samples. Shsy5y cells may be transformed to various types of functioning neurons by adding particular substances, making it a model for neurodegenerative illnesses. In addition, the Shsy5y cell line has been widely employed in experimental neurological investigations, including analysis of neuronal development, metabolism, and function in relation to neurodegenerative processes, neurotoxicity, and neuroprotection [37]. Astrocytes are a type of glial cell that outnumber neurons by a factor of five. They tile the entire central nervous system (CNS) and perform a variety of important complicated tasks in a healthy CNS [38]. From the samples in SCIS, the mean annotation count per image is 34, 337, and 80 for Cort, Shsy5y, and Astro, respectively. The mean mask area is 240, 224, and 906 (number of pixels) for Cort, Shsy5y, and Astro, respectively. It is observed that Astro cells are significantly larger than the other two. The Cort cells, despite the small size, are also sparse. The Shsy5y cells present the highest the density among the three cell lines. In addition to the 606 training samples provided to the public, there were 243 samples in the test set. Teams do not have access to the annotations of all samples in the test set. Prediction results need to be put into an spreadsheet, which is submitted to the competition sever for score calculation. All images in the dataset have the same scale, 520 × 704.

Each sample in SCIS is a PCM image with a number of mask annotations. Figure 2 shows three samples (one cell line for each) in the SCIS dataset. For each sample, we display the original PCM image (left) and the image with human-annotated masks (right). This visualization confirms

1https://www.kaggle.com/c/sartorius-cell-instance-segmentation
FIGURE 2: Three samples from the SCIS dataset with one sample for each cell line. For each sample, we display the original PCM image (left) and the human-annotated segmentation masks (right). The subfigures include: (a) a Cort sample, (b) a Shsy5y sample, and (c) an Astro sample.

TABLE 1: Stats of the three cell lines in SCIS

| Type | # samples | Annotation count | Mask area (# pixels) |
|------|-----------|------------------|----------------------|
| Cort | 320       | 34 4 108         | 240 33 2,054        |
| Shsy5y | 155      | 337 49 790      | 224 30 2,254        |
| Astro | 131      | 80 5 594         | 906 37 13,327       |

the density and size of the three cell lines. Also, we notice that the Cort cells are more regular and round-shaped, while the Astro cells are the most irregular and star-shaped. The morphology diversity is the main challenge for accurate instance segmentation for neuronal cells.

B. SYSTEM OVERVIEW

Figure 3 shows an overview of the proposed learning framework. We first fine-tune a Mask RCNN model on the LIVE-Cell dataset. The weights of this pre-trained model is used to initialize the main model, which is fine-tuned on enhanced SCIS dataset, where each original image has been enhanced with adjusted contrast. It is noted that the Mask RCNN models used for fine-tuning adopt the PAPF-PAN structure as their backbone networks for feature extraction.

C. IMAGE ENHANCEMENT

To enhance the contrast of the PCM images, we have adopted a contrast enhancement algorithm developed in [39]. The core design idea of this algorithm is to enhance the under-exposed areas of an image and keep the well-exposed areas preserved. At a high level, the algorithm can be broken down into four steps. First, a weight matrix is designed via illumination estimation for image fusion. Second, a camera response model is employed to generate a synthetic image.
FIGURE 3: System overview of the proposed learning pipeline. The PAFA module is integrated into the backbone of the original Mask RCNN model. The PAFA-enabled Mask RCNN model is firstly pre-trained on the LIVECell dataset, then fine-tuned on the enhanced SCIS training set, and evaluated on the hidden test set.

FIGURE 4: A four-step algorithm for contrast enhancement: 1) a weight matrix \( W \) is determined via illumination estimation; 2) a camera model is utilized to produce a synthetic image via a brightness transform function \( g(P, k) \) that maps the input \( P \) with an exposure ratio \( k \) to produce \( P' \); 3) \( P \) and \( P' \) are scaled by \( W \) and \( 1 - W \) via element-wise multiplication, respectively; 4) outputs of step (3) are fused via summation to produce the enhanced result \( R \).

with a different exposure. Next, the best exposure ratio is determined to enhance the contrast of regions that are not well-exposed in the original image. Lastly, the original input and the synthetic image are fused via the weight matrix to produce the enhanced image. Figure 4 describes the process of contrast enhancement. In this figure, \( P \) denotes the original image; \( W \) refers to the weight matrix; \( P' \) refers to the synthetic image obtained from a brightness transform function \( g(P, k) \) that maps the input \( P \) with an exposure ratio \( k \) to produce \( P' \). The final enhanced image is obtained via the following equation.

\[
R^c = W \odot P^c + (1 - W) \odot g(P^c, k)
\]

where \( c \) denotes the color channel and \( \odot \) refers to an element-wise multiplication operation. For a detailed procedure for the determination of \( W, g, k \), we refer readers to [39].

D. MASK RCNN

Mask RCNN was developed on top of the Faster RCNN architecture; the latter was used for the object detection task, while Mask RCNN has a third branch used for instance segmentation, in addition to the two existing branches in Faster RCNN used for class prediction and bounding box regression. At a high level, Mask RCNN consists of two stages. In stage I, a region proposal network (RPN) slides over the multi-scale feature maps produced from the backbone network (i.e., PAPF-PAN in our case) and generates a collection of region proposals that may contain objects of interest. The RPN is essentially a lightweight neural network with a classifier and a regressor. The former predicts whether or not a proposal having the target object, and the latter regresses the corresponding coordinates of the proposed regions. For each point in a feature map, a total of nine proposals are considered. Thus, for an \( H \times W \) feature map,
a total of $9HW$ proposals are generated. In the second stage, the proposed regions are fed through a region of interest (RoI) Align module to properly rescale these region proposals using bilinear interpolation rather than quantization, which was used by RoI Pooling in Faster RCNN and could lead to loss of critical information in the segmentation task due to mis-alignments between the extracted features and RoI. After RoI Align, all region proposals have the same scale and are then fed into three separate branches for class prediction, bounding box regression, and segmentation mask prediction.  

The loss function of Mask RCNN can then be defined as $L = L_{\text{mask}} + L_{\text{cls}} + L_{\text{cord}}$, where $L_{\text{cls}}$ and $L_{\text{cord}}$ stands for classification and coordinate losses, which are the same as the ones used in Faster RCNN [24]; $L_{\text{mask}}$, on the other hand, is defined as the average binary cross-entropy loss for $K$ classes. For our task, there are only foreground (neuronal cell) and background pixels, namely, $K = 2$. Therefore, the mask head outputs a $2 \times m \times m$ tensor for each RoI; the tensor goes through a per-pixel sigmoid function to facilitate the calculation of $L_{\text{mask}}$. 

**E. PAPF-PAN**

Figure 5 shows the proposed PAPF module, taking the $1/4$ and $1/8$ feature map pair as an example input. First, the $H/4 \times W/4$ feature map, denoted by $F_l$ where $l$ refers to its current layer, goes through a downsampling module, reducing its scale to match the dimension of the $H/8 \times W/8$ feature map, denoted by $F_{l-1}$ in the lower level. The following equations describe a detailed process of the PAPF module.

$$F'_{l-1} = \text{downsample}(F_l) \quad (2)$$
$$W_{PA} = \sigma(F'_{l-1}) \quad (3)$$
$$F''_{l-1} = [F_{l-1}; F'_{l-1}; W_{PA} \odot F_{l-1}] \quad (4)$$

in which $\sigma$ represents a sigmoid function that converts values of a feature map into the range of 0 and 1, $W_{PA}$ represents the pointwise attention tensor, $\cdot\cdot$ denotes the concatenation operation, and $\odot$ denotes the element-wise multiplication. $W_{PA}$ consists of the weights that are learned during training, and each weight value represents the importance of a point in the feature map. The output of the PAPF module, $F''_{l-1}$, is a concatenation of $F_{l-1}$, $F'_{l-1}$, and $W_{PA} \odot F_{l-1}$, which take into account both cross-scale feature maps and the importance of the individual points in the lower level feature map.

The original Mask RCNN neural architecture has experimented with the FPN backbone, which is the predecessor of PAN. We integrate the proposed PAPF module into the PAN architecture, which is utilized as the backbone for feature extraction. Figure 6 showcases the PAPF-PAN neural architecture, in which the PAPF modules are marked in red and integrated into the downsampling path to add cross-scale attention to the architecture.

**F. TRANSFER LEARNING**

Transfer learning refers to a class of methods for transferring knowledge learned in one domain or task to other domains or task scenarios [40]. For models performing different types of tasks, transfer learning can also be used to improve training and accelerate convergence. It should be noted that when the feature distributions between domains are very different, knowledge is not effectively transferred. In other words, transferring knowledge between two semantically similar domains is easier and more effectively than the transfer between two dissimilar or even unrelated domains. Therefore, we need to evaluate the differences between the domains before transfer learning can be performed.

For our task, we choose LIVECell as the source domain to obtain a pre-trained model. Despite the differences of cell lines between LIVECell and SCIS, a large number of low-level and intermediate-level features are shared across the two domains. Therefore, LIVECell serves as a decent domain to learn these features that can be effectively transferred to the target domain. Also, LIVECell eight times the size of SCIS, offering abundant semantic features of cells to learn and transfer. We take the Mask RCNN model pre-trained on the Microsoft COCO dataset [41] and modify the last layer in each detection head to fit our task’s output classes, namely, background and cell, rather than the 80 classes in COCO. Two steps of fine-tuning is performed. First, we fine-tune the model on the LIVECell dataset, followed by a further fine-tuning on the SCIS dataset. The details of the transfer learning configuration is reported in the next section.

**IV. EXPERIMENTS AND RESULTS**

In this section, we report the details of the experimental design and key results with our analysis and insights.

**A. METRIC**

The mean average precision at multiple intersection over union (IoU) thresholds is used to evaluate this competition. The IoU of a proposed set of object pixels vs a true set of object pixels is determined as follows:

$$\text{IoU}(A, B) = \frac{A \cap B}{A \cup B} \quad (5)$$

The metric sweeps across a range of IoU thresholds, calculating an average precision value at each point. The threshold values range from 0.5 to 0.95, with a 0.05 step size: (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95). In other words, a predicted object is considered a “hit” if its intersection over union with a ground truth object is greater than 0.5 at a threshold of 0.5.

The number of true positives (TP), false negatives (FN), and false positives (FP) arising from comparing the predicted object to all ground truth objects is used to compute a precision value ($P_{re}$) for each threshold value $t$:

$$P_{re} = \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \quad (6)$$
FIGURE 5: The proposed PAPF Module. Consider a feature map, say, the one with a scale $H/4 \times W/4$, is downsampled and normalized via a sigmoid function to obtain a tensor of size $H/8 \times W/8$, which is a learnable pointwise attention tensor. Element-wise multiplication is applied on the attention tensor and the original $H/8 \times W/8$ feature map within the aggregation path. Lastly, a concatenation operation is used to fuse three feature maps, including the one with cross-layer attention, the one from the current layer, and the one from the previous layer in the path.

When a single predicted object matches a ground truth object with an IoU over the threshold, it is considered a true positive. A false positive shows that there was no ground truth object associated with a predicted object. A false negative means that there was no associated predicted object with a ground truth object. The mean of the above precision values at each IoU threshold is then used to calculate the average precision (AP) of a single image:

$$AP = \frac{1}{|\text{thresholds}|} \sum_t P_{Re_t}$$  

(7)

Lastly, the score returned by the competition metric is the mean taken over the individual average precisions of each image in the test dataset.

**B. BASELINES**

As of writing, there were no peer reviewed articles that utilize the same dataset. Therefore, we chose the top three-ranked solutions from the Kaggle competition as the baselines. The competition host divided the test set into a public (including 41% of the test data) and a private set (59% of the test data). During the competition, participants can only view their ranks and scores on the public leaderboard (LB). The private LB is not visible to participants until the competition has concluded. The final places were determined based on the private LB scores. For this competition, the the places of the top three solutions stay the same for both public and private LBs. For a fair comparison, we also report our scores on both LBs for a fair and complete comparison. A brief description of these solutions is provided below.

- 1st place [1] Takuoko and Tascj adopted a two-stage learning pipeline that utilized three YOLOX [42] models for BBox prediction. The predicted BBoxes are then fed through two Mask RCNN heads and four UperNet [43] heads for the mask detection. The final mask prediction is a simple mean of the predictions of all heads. It is noted that both BBox and mask ensembles have been utilized to boost performance.

- 2nd place [2] NVNN and sheep developed a framework that is an ensemble of two object detection models, one UNet, and two MaskRCNN models. For object detection, the team utilized yolov5x6 and effdetD3. Weighted box fusion [44] was adopted for BBox optimization.

- 3rd place [3] Chenevert et al. provided an ensemble solution that integrates Mask RCNN and UNet to generate the binary masks that are grouped and averaged to produce the final masks. Pseudolabels are adopted to enhance the quantity and diversity of the training data.

All top three teams utilized ensemble models that are pre-trained on LIVECell. Compared to these solutions, our proposed method does not rely on ensemble methods and has achieved comparable performance. Results are presented and described in section IV-E.

**C. FINE-TUNING CONFIGURATION**

The proposed models are implemented using Python 3.8 and PyTorch 1.8.0. Experiments were conducted on a Windows
workstation with an i7-10875h CPU, and a GTX2080TI GPU was used for acceleration. We chose the Adam optimizer with a learning rate of 0.0001, beta1 and beta2 being 0.9 and 0.999, respectively. Also, the eps parameter is set to be 1e-08 to prevent the denominator from being 0. The total fine-tuning epochs is set to be 5,000. The IoU threshold that yields the highest score is 0.46. When performing transfer learning, considering the variability between different datasets, we first adopted the warmup strategy, using a learning rate of 0.0001, and want to perform a subtle transfer of the data, driving it to gradually change from LiveCell to adapt to the SCIS, with the warmup epochs number set to 200. The learning rate then linearly grows from its base value. The dataset has been split into a training and validation set in the ratio of 8:2, resulting in a training set of 485 samples (256 Cort, 124 Shsy5y, and 105 Astro) and a validation set of 121 samples (64 Cort, 31 Shsy5y, and 26 Astro).

D. IMAGE ENHANCEMENT

We have applied the image contrast enhancement algorithm described in Section III.C to all images in the SCIS dataset. Figure 7 shows two examples, where subfigures (a) and (d) are two original images from the dataset, (b) and (e) are the enhanced images, (c) and (f) show the pixel-level difference between the original and enhanced images for (a) and (d), respectively. It is observed that there is no significant visual difference that can be captured by human eyes between the original and enhanced images, but the differences do exist and can be clearly shown in subfigures (c) and (f), meaning that the contrast for both images has been enhanced. Further more, our experimental results show that the mAP improves when the model is fine-tuned on the enhanced images.

E. RESULTS AND ANALYSIS

We report the mAP scores in Table 2 which also shows the result of an ablation study. It is observed that as the three boosters are incrementally integrated into the learning model, the public score increases from 0.238 to 0.291, 0.318, and 0.341, showing a combined gain of 10.3%, compared to the base model $M_1$. Similarly, the private score increases from 0.247 to 0.355, posting a gain of 10.8%. The similarity between LIVECell and SCIS is beneficial to the prediction accuracy. We also notice that the addition of transfer learning on LIVECell has improved the public and private mAPs by 5.3% and 5.1%, respectively, as we compare models $M_1$ and $M_2$. It is noted that $M_1$ has been pre-trained on the COCO dataset that contains 80 classes (e.g., person, dog, bike, and chair) that are very dissimilar to the cells, meaning that the patterns learned from COCO may not be useful when transferred to SCIS. After fine-tuning $M_1$ on the LIVECell, we obtain $M_2$, which presented a higher mAP, meaning that the patterns learned from LIVECell are more meaningful and informative for the cell segmentation task. The improvement is due to the similarity between the LIVECell and SCIS datasets. In addition, contrast enhancement has driven a performance gain of 2.7% and 2.3% on the public and private LBs, respectively, meaning that contrast plays a crucial role in detection accuracy. The contrast enhancement algorithm used in our method can effectively improve the under-exposed areas without affecting the well-exposed...
areas. Lastly, the integration of the PAPF module improves the score by 3.2% and 3.4% on the public and private LBs, respectively. The gain indicates that the proposed cross-scale attention mechanism allows the model to better identify the critical areas with more informative patterns during training.

Also, our best model, $M_4$, shows similar performance compared to the top three ranked solutions. Specifically, $M_4$ outperforms the second and third places in both public and private LBs. Compared to the first place, $M_4$ presents the same public score and is only 0.1% worse in the private score. Most importantly, our solution is based on a single model, while all three top solutions rely on ensemble methods that aggregate the predictive power of multiple models.

In addition to mAP, the time duration was 2 seconds per epoch (including 606 images) for training and around 0.08 second per epoch (including 121 images) for inference. In other words, the processing speed for inference is about five times faster than the training speed. This is reasonable since the GPU has to handle both forward and backward propagation during training, while only forward propagation is needed during inference.

Some qualitative results are also reported in Figure 8. Subfigures (a)-(d) represent the original image (an Astro sample), and the image with predicted masks and bounding boxes for the models $M_2$ through $M_4$. Although the task only requires segmentation masks, our Mask R-CNN-based model can generate a mask and an associated bounding box for each detected cell instance. Several detections in (c) and (d) are numbered for the purpose of illustration. It is visually observed that $M_3$ and $M_4$ can detect more cell instances than $M_2$, since more boosters are applied on $M_3$ and $M_4$. Also, $M_4$ is more accurate than $M_3$. For example, for the instances marked with 1, 2, 3, 4, 5, 7, and 11 in subfigure (d), $M_4$ provides more accurate detection, and $M_3$ either misses certain areas of cells or the whole cell body, or splits a cell apart. Meanwhile, the detections marked with 6, 7, 8, 9, and 10 in subfigure (c) are more or less the false positives generated by $M_3$. This typical individual sample shows the efficacy of the three boosters combined to drive a performance gain.

V. DISCUSSION

DNN-based neuronal cell instance segmentation has been an crucial technique that could help uncover mysteries of cellular mechanisms in neural science. Current approaches fall short in segmentation precision due to various challenges, such as irregular shapes, low contrast, and cell overlapping and clustering. We propose a DNN-based learning framework with three performance boosters to address the aforementioned challenges. Experiments show that the joint effect of these boosters has been promising with a combined gain in mAP of 10.3%. Our method also has also outperformed two strong baselines by 6.8% and 3.4%.

This work has the following limitations that also suggest
TABLE 2: mAP comparison. Abbreviations in the table include Transfer Learning (T.L.), Image Contrast Enhancement (I.C.E.), and Ensemble Method (E.M.)

| Model                | T.L. on LIVECell | I.C.E. | PAPF-PAN | E.M. | Public Score | Private Score |
|----------------------|------------------|--------|----------|------|--------------|---------------|
| M1                   | ×                | ×      | ×        | ×    | 0.238        | 0.247         |
| M2                   | ✓                | ×      | ×        | ×    | 0.291        | 0.298         |
| M3                   | ✓                | ✓      | ✓        | ×    | 0.318        | 0.321         |
| M4                   | ✓                | ✓      | ✓        | ×    | 0.350        | 0.355         |
| Takuoko and Tasci (1st place) | ✓  | NA    | NA       | ✓    | 0.350        | 0.356         |
| NVNN and sheep (2nd place) | ✓  | NA    | NA       | ✓    | 0.347        | 0.354         |
| Chenevert et al. (3rd place) | ✓  | NA    | NA       | ✓    | 0.346        | 0.354         |

FIGURE 8: (a)-(d) represent the original image (an Astro sample), and the images with predicted masks and bounding boxes for the models M2 through M4, respectively.

our future research directions. First, our model is largely based on Mask RCNN, while there are many other models that have been designed for instance segmentation. It is desired to explore a wider range of model options since the three boosters can be readily integrated. Second, in this study we did not explore that how cell types can be utilized to improve the model performance, while Chenevert’s solution has adopted a simple yet effective strategy to decide the number of masks to be kept in an inference result based on the predicted cell line. It would be interesting to explore more cell line-dependent features to enhance the overall segmentation performance. Lastly, the main challenges of cell instance segmentation are the irregularity and depression of cell morphology, cell aggregation and adhesion, and the background noise in the PCM images. Existing studies, including the proposed method, lack a quantitative perspective on how each aspect of these challenges can be improved. In other words, it would be interesting to find out how an overall mAP gain brings a quantified improvement on each challenge. This type of empirical study has not been seen in the literature and will be insightful.

ACKNOWLEDGMENT
We thank the anonymous reviewers for their careful reading of our manuscript and their insightful comments, which are essential to the improvement of the manuscript quality.

REFERENCES
[1] W. Halfter, R. Chiquet-Ehrismann, and R. P. Tucker, “The effect of tenascin and embryonic basal lamina on the behavior and morphology of neural crest cells in vitro,” Developmental Biology, vol. 132, no. 1, pp. 14–25, Mar. 1989.
[2] S. K. Finley and M. F. Kritzer, “Immunoreactivity for intracellular androgen receptors in identified subpopulations of neurons, astrocytes and oligodendrocytes in primate prefrontal cortex,” Journal of Neurobiology, vol. 40, no. 4, pp. 446–457, 1999.
[3] O. Lindvall and Z. Kokaia, “Stem cells for the treatment of neurological disorders,” Nature, vol. 441, no. 7097, pp. 1094–1096, Jun. 2006.
[4] W. H. Organization, Neurological Disorders: Public Health Challenges. World Health Organization, 2006.
[5] P. J. Keller and M. B. Ahrens, “Visualizing Whole-Brain Activity and Development at the Single-Cell Level Using Light-Sheet Microscopy,” Neuron, vol. 85, no. 3, pp. 462–483, Feb. 2015.
[6] M. R. Arvinz, K. G. Larkin, C. J. R. Shoppard, N. I. Smith, and C. J. Cogswell, “Linear phase imaging using differential interference contrast microscopy,” Journal of Microscopy, vol. 214, no. 1, pp. 7–12, 2004.
[7] C. R. Burch and J. P. P. Stock, “Phase-Contrast Microscopy,” Journal of Scientific Instruments, vol. 19, no. 5, pp. 71–75, May 1942.
[8] C. Edlund, T. R. Jackson, N. Khalid, N. Bevan, T. Dale, A. Dengel, S. Ahmed, J. Trygg, and R. Sjögren, “LIVECell—a large-scale dataset for label-free live cell segmentation,” Nature Methods, vol. 18, no. 9, pp. 1038–1045, Sep. 2021.
[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015.
[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
[11] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object Detection With Deep Learning: A Review,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.
[12] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, “Deep learning in video multi-object tracking: A survey,” Neurocomputing, vol. 381, pp. 61–88, Mar. 2020.
[13] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Review,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2021.
[14] E. Shelleher, J. Long, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[15] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[16] Y. Yuan, X. Chen, and J. Wang, “Object-Contextual Representations for Semantic Segmentation,” in Computer Vision – ECCV 2020, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M.
