

Kaggle Neuronal Cells Instance

Segmentation Top 1% Solution

Xinyue Wang

Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, USA

ABSTRACT

Neurological disorders have caused increasing death and disability across the globe, which increases the demand for corresponding treatment and rehabilitation. As a key step in neurological disorders drug development, neuronal cells segmentation in light microscopy is vital to obtain deadly disorders' response to treatment. However, common manual neuronal cells instance segmentation is inefficient, time-intensive, and labor-costing. To automatically and accurately segment neuronal cells from images, I developed a neuronal cells instance segmentation system, including an instance segmentation model based on Cascade Mask R-CNN X152 and a customized post-processing pipeline. The resulting robust segmentation system leverages several techniques that achieved 33.2% mAP, leading to the top 1% place on Kaggle Sartorius Cell Instance Segmentation Competition. This work potentially provides an efficient and effective alternative for manually segmenting neuronal cells, which saves labor costs and further facilitates drug development in curing neurological disorders.

Keywords: Neurological disorders, Cells instance segmentation, Post-processing

INTRODUCTION

As a leading cause of death and disability across the globe, neurological disorders, including neurodegenerative diseases such as Alzheimer's and brain tumors, have caused an increasing demand for treatment, rehabilitation, and support services for neurological disorders[1]. However, the difficulty of quantifying the response of these deadly disorders to treatment is often preventing researchers from making progress. A widely accepted means is observing the neuronal cells change via light microscopy, which is straightforward and non-invasive[2]. Note that the large number and tiny size of neuronal cells in single light microscopy, it is a time-consuming and challenging task for physicians and radiologists to segment each neuronal cell and do analysis on it. To save labor costs and prevent medical resources waste, many computer vision methods were proposed and proved to be potential help for neuronal cells instance segmentation, which is likely to facilitate the drug development for disorders treatment [3, 4, 5].

Among those previous CV assistance research, a well-known and successful one is the work of Dr. Edlund and his team [6]. They developed an instance segmentation model based on their newly collected neuronal cells light microscopy data set called LIVECell. Although this model has achieved high accuracy on most kinds of neuronal cells instance segmentation, it fails to handle particular neuronal cells like the neuroblastoma cell line(sh-sy5y), since they have irregular and concave morphology, which can not be recognized well by the common mask head [6]. To improve the segmentation performance for these particular neuronal cells, the competition host, Sartorius, a life science company has collected new neuronal cells light microscopy data set and hosted

this neuronal cells instance segmentation competition on Kaggle to find more robust segmentation solutions. The competition data set is consists of three types of neuronal cells which are cort, sh-sy5y, astro, while only the sh-sy5y cell data is collected in the LIVECell data set.

In this project, my main work is to build an automated pipeline based on deep learning knowledge to segment the individual cell part in light microscopy. To better segment these irregular shape neuronal cells, I adopted the Cascade Mask R-CNN model to acquire high-quality masks, and it was proved to be a powerful structure that has nontrivial improvements over the previous state-of-the-art, Mask R-CNN [7]. In addition, I designed and implemented a customized post-processing pipeline to refine different neuronal cells segmentation masks respectively, which brings a significant increase in segmentation performance. My contribution can be described as follow:

1. Develop a robust and effective neuronal cells instance segmentation pipeline on a new data set. Currently achieved 33.2% mAP(mean Average Precision), ranking at 21/1457, which is among the top 1% on the public leaderboard.
2. Provide effective methods and ideas for mask post-processing after instance mask prediction, which denotes that the other data analysis part of instance segmentation except for segmentation network are equally important and worth doing research.

METHODS

In this part, my work can be divided into three general steps, model training, model inference, and segmentation post-processing, while the data preprocessing part in the model training will be illustrated as a single subsection. The well-trained model after

the first step will be able to do neuronal instance segmentation, and by adopting the segmentation post-processing, the refined segmentation result would be more accurate.

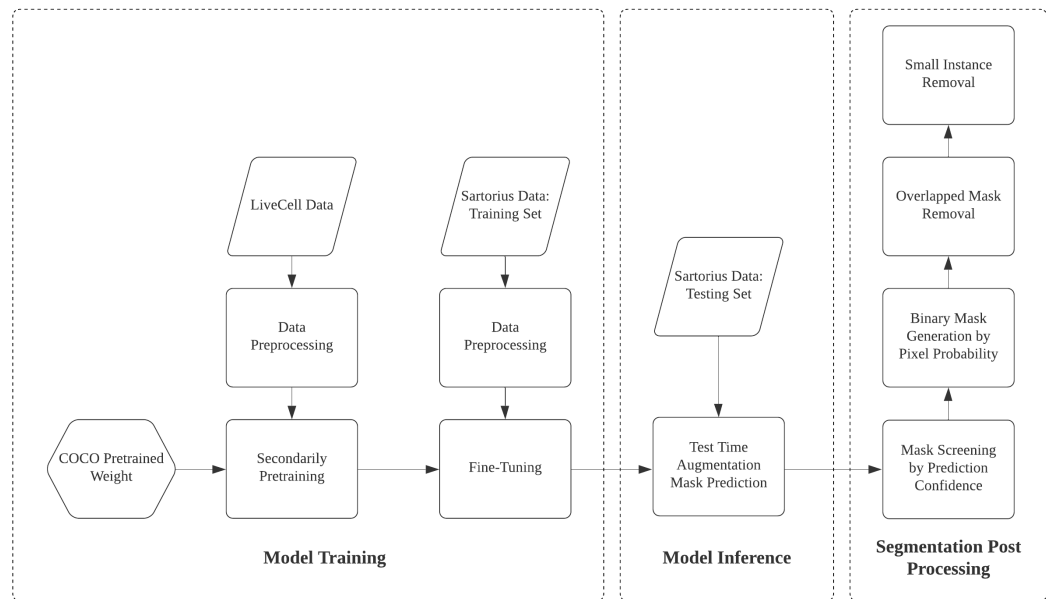


Figure 1. Project Workflow

Data Preprocessing

The competition data includes 606 labeled light microscopy images as the training set, 1972 unlabeled light microscopy images as semi-supervised data, whose width and height are all 704 and 520. In this part, I only utilized and the utility of semi-supervised data will be discussed in the discussion. 5 folds of the training set and validation set were generated to do cross-validation, where the cell class percentage were remained the same in every fold to preserve the balance of the sample between different classes. Here I noticed that the number of training samples is small and not sufficient to retrain a big capacity network and neuronal cells image is not common in most visual tasks, which makes directly utilizing the common objects pretrained weight of other big capacity models not ideal. Therefore, I chose the LIVECell neuronal cells light microscopy data

set including 7 kinds of cells and 5239 images in total to do secondarily pretraining based on the pretrained weight which was trained on the COCO(Common Objects in Context) data set. The reasons why still adopted common objects pretrained weight as a starter instead of fully fresh retraining are that the basic details like textures of daily objects image still have some commonalities with neurons image, and starting from pretrained model often outperforms training from scratch [8]. To be more specific, Only the LIVECell data set is used for secondarily pretraining, and the competition training set is all the data I utilized in fine-tuning for the secondarily pretrained model.

After collecting data, I applied different data augmentation techniques on respectively. Multi-scale training is an effective way to improve generalization and better detect objects of various sizes [9], therefore I consistently utilize multi-scale training in my pretraining and fine-tuning process. For the pretraining data set, multi-scale resize transformation including (1056, 780), (1232, 910), (1408, 1040), and both horizontal and vertical flip were applied. For the fine-tuning data set, multi-scale resize transformation including (1408, 1040), (1490, 1100) and the same two kinds of flip were adopted. Besides, random rotation and some random small range brightness change were conducted to make the model has better generalization. Since the cort cell and sh-sy5y cell were small, all the input sizes were resized larger instead of using the original size(704*520) to better capture their features. All the transformations were experimented with and selected carefully to keep the augmentation image was still in the original data distribution.

Model Training

In this part, after some experiments, I found bigger and deeper models have superior advantages than the smaller ones. Also, to effectively decrease false positives and obtain more high-quality masks in different levels of IoU(Intersection of Union) when calculating the mean average precision, cascade Mask R-CNN X152 was decided as the base network. The ResNeXt with a depth of 152 layers was used as the backbone in this network. Compared to the default settings in Cascade Mask R-CNN X152, smaller anchor sizes(8, 16, 32, 64, 128) were set to better extract the visual information of small cells.

With imported COCO pretrained weight, the base network was secondarily pretrained using SGD optimizer with the smooth L1 loss for 200000 iterations. In the fine-tuning part, the model trained in the previous step was fine-tuned with the same SGD optimizer and smooth L1 loss for 5000 iterations, while the learning rate is smaller than the one used in pretraining. During the pretraining and fine-tuning period, various training techniques including weight decay, multi-step scheduler, and early-stopping to boost the model's better convergence and prevent over-fitting.

Model Inference

This part is simple and similar to most instance segmentation inference processes. Test time augmentation is proved as a stable and powerful augmentation strategy to significantly increase prediction accuracy in cell segmentation [10]. To be more specific, for a single test image, the model would generate different predictions corresponding to different way augmented test images and average these predictions as the final inference

result of this test image. In this part, a multi-scale augmentation including (1490, 1100) and (1408, 1040) and horizontal flip was applied at the same time, as a result, the final inference result is the ensemble of 4 augmented predictions. The predicted mask here were kept as pixel probability instead of a binary mask for the following post-processing.

Segmentation Post Processing

Note that different cells have their morphology characteristics, it is a need to do customized post-processing for particular cells. The post-processing pipeline consists of four steps, Mask Screening, Binary Mask Generation, Overlapped Mask Removal, and Small Instance Removal.

Firstly, to remove low confidence false positives, generated masks were screened by their prediction confidence based on the tuned class-wise confidence threshold. Then, since the individual probability of each pixel being mask pixel in different cell classes is different, the rest of the high confidence probability masks were converted to binary masks based on tuned class-wise pixel probability threshold, which is reasonable and ought to be effective. Because for the high confidence model, a low mask probability pixel is also likely to be a mask pixel and for the weak model, only a high mask probability pixel is probable to be the mask pixel. The third step is to remove the overlapped mask by the ranking of prediction confidence. The higher confidence mask would have priority to occupy the overlapped pixels, and this overlapped part of pixels would be correspondingly removed from the lower confidence mask. To clean up the small outlier and some severely broken masks caused by overlapped mask removal, the small instance mask were removed based on the tuned class-wise mask area threshold due to obvious cell size differences of different cell classes.

EXPERIMENTS

In the part of the experiment, the model performance was evaluated based on 5 folds cross-validation, while the evaluation metric is mAP. As the equations below, mAP is computed as the mean of the above precision values at each IoU threshold, where IoU is defined as an intersection over the union between the proposed set of object pixels and a set of true object pixels. Note that IoU thresholds are (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95).

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

$$mAP = \frac{1}{|thresholds|} \sum_t \frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

The model overall performance experiments results are listed as follows, the public baseline is a public solution that is developed based on Mask R-CNN. The results show my solution has a significant advantage compared to the public baseline and the post-processing pipeline improved overall performance by about 1% mAP whether on cross-validation results or on the public leaderboard, which could be regarded as an unseen test data set.

Model	Validation Set	Public Leaderboard
Public Baseline	27.8% (mAP)	29.4%(mAP)
Naive Model Inference (mine)	31.0% (mAP)	32.0% (mAP)
Model Inference with Post Processing (mine)	32.0% (mAP)	33.2% (mAP)

Table 1. Model overall performance comparison

The respective cell results are listed as follows, the results proved that a customized post-processing pipeline is effective and beneficial for improving the segmentation

performance of all three cell classes respectively, as a result, boosting overall prediction accuracy. Compared to the public baseline, the prediction performance on sh-sy5y and astro cells has been increased by up to 10 % and the prediction accuracy on cort cell is also getting nontrivial enhancement.

Model Inference(Validation Set)	Sh-sy5y	Astro	Cort
Public Baseline	14.1% (mAP)	12.6% (mAP)	37.2%(mAP)
Naive Model Inference (mine)	23.8% (mAP)	19.6% (mAP)	38.9%(mAP)
Inference with Post Processing (mine)	24.6% (mAP)	20.4% (mAP)	40.1% (mAP)

Table 2. Model performance comparison on individual cell class

From figure 2, it is clear that most neuronal cells are segmented and the refined mask has high IoU with the ground truth instance. It also explains why Cort cell class has the best performance, which is because of its relatively regular morphology and a small number. The segmentation performance of sh-sy5y and Astro are not that satisfied because they are either many and varied or large sizes, in addition to their irregular shape. Due to the competition host requirement that the prediction submission must have no overlapped mask, some accurate predictions are broken, thereby decreasing mAP, which can be told from visualization.

DISCUSSION

In this project, many strategies including other powerful networks and more complicated data augmentation got worse performance than they are supposed to get. For instance, Mask R-CNN with ResNeSt as backbone was hard to converge and always perform much worse than Mask R-CNN with ResNet or ResNeXt as the backbone, while the LIVECell team achieved good results using this backbone [6]. As the newest state-of-the-art in instance segmentation, the recently hit network, Swin Transformer [11] was

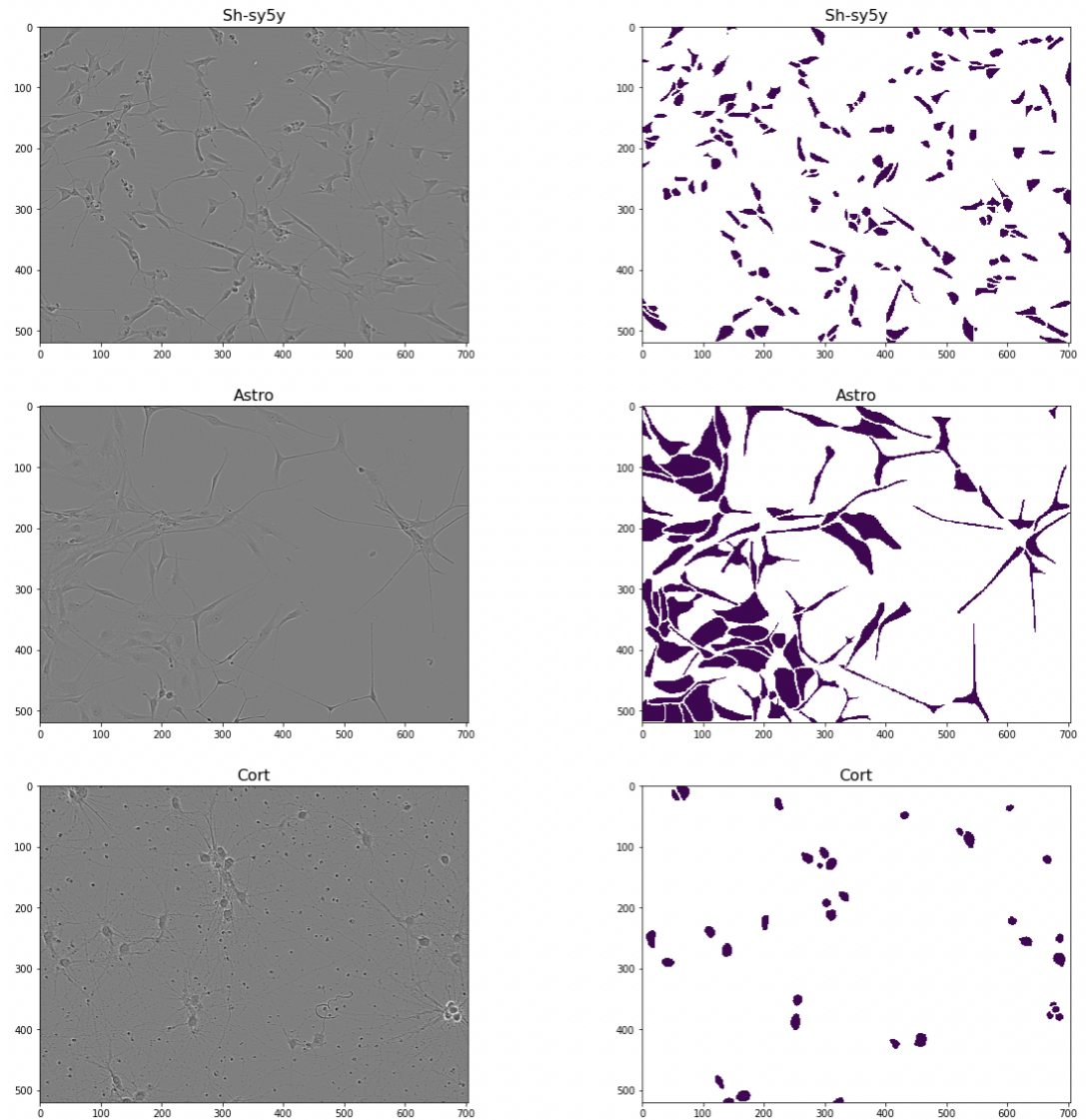


Figure 2. segmentation Mask of Different cell class

also not doing as well as Cascade R-CNN in this task. These differences could be caused by the data distribution variation between the competition data set and the LIVECell data set. In addition, the negative influence brought by complicated data augmentation such as colorful augmentation could be explained as the induced disruption to the original data distribution.

It is also clear that my method still has some limitations, and a lot of work needed to be done in future research. Firstly, there is still room for the improvement of model

generalization. The utility of pseudo labels in semi-supervised data is likely to be helpful [12], which is a priority in future work. Secondly, the competition data itself has contained some low-quality labels, and it might introduce too much noise when the model is training. It could be solved by doing more detailed data cleaning like removing high prediction confidence but low accuracy sample and manually refining these labels. Until now, there is only one model used in the segmentation pipeline, since the regular ensemble method such as vote and average would bring more false positives than one single model. However, a customized ensemble strategy is highly probable to increase overall performance by some special ensemble strategies such as weighted mask fusion, which needs to be designed and implemented in the next step.

CONCLUSION

Neurological disorders have appealed to more and more concerns because of the death and disability they cause. As a key step to disorders drug development, neuronal cells segmentation in light microscopy is vital to obtain deadly disorders' response to treatment. However, common manual neural cells instance segmentation is inefficient, time-intensive, and labor-costing. In this paper, I presented my work on Kaggle Sartorius Cell Instance Segmentation Competition, a neuronal cells instance segmentation system, including an instance segmentation model based on Cascade Mask R-CNN X152 and a customized post-processing pipeline. The resulting robust segmentation system leverages several techniques that achieved 33.2% mAP, leading to the top 1% place among all the competitors. This work potentially provides efficient and effective alternatives for manually segmenting neuronal cells, which saves labor costs and further

facilitates drug development in curing neurological disorders.

REFERENCES

- [1] Feigin Valery L, Nichols Emma, Alam Tahiya, et al. Global, regional, and national burden of neurological disorders, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016 *The Lancet Neurology*. 2019;18:459–480.
- [2] Wilt Brian A, Burns Laurie D, Wei Ho Eric Tatt, Ghosh Kunal K, Mukamel Eran A, Schnitzer Mark J. Advances in light microscopy for neuroscience *Annual review of neuroscience*. 2009;32:435–506.
- [3] Ronneberger Olaf, Fischer Philipp, Brox Thomas. U-net: Convolutional networks for biomedical image segmentation in *International Conference on Medical image computing and computer-assisted intervention*:234–241Springer 2015.
- [4] Yi Jingru, Wu Pengxiang, Jiang Menglin, Huang Qiaoying, Hoepfner Daniel J, Metaxas Dimitris N. Attentive neural cell instance segmentation *Medical image analysis*. 2019;55:228–240.
- [5] Stringer Carsen, Wang Tim, Michaelos Michalis, Pachitariu Marius. Cellpose: a generalist algorithm for cellular segmentation *Nature Methods*. 2021;18:100–106.
- [6] Edlund Christoffer, Jackson Timothy R, Khalid Nabeel, et al. LIVECell—A large-scale dataset for label-free live cell segmentation *Nature methods*. 2021;18:1038–1045.

- [7] Cai Zhaowei, Vasconcelos Nuno. Cascade r-cnn: High quality object detection and instance segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2019.
- [8] Tajbakhsh Nima, Shin Jae Y, Gurudu Suryakanth R, et al. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*. 2016;35:1299–1312.
- [9] Felzenszwalb Pedro, McAllester David, Ramanan Deva. A discriminatively trained, multiscale, deformable part model in *2008 IEEE conference on computer vision and pattern recognition*:1–8Ieee 2008.
- [10] Moshkov Nikita, Mathe Botond, Kertesz-Farkas Attila, Hollandi Reka, Horvath Peter. Test-time augmentation for deep learning-based cell segmentation on microscopy images *Scientific reports*. 2020;10:1–7.
- [11] Liu Ze, Lin Yutong, Cao Yue, et al. Swin transformer: Hierarchical vision transformer using shifted windows *arXiv preprint arXiv:2103.14030*. 2021.
- [12] Chapelle Olivier, Scholkopf Bernhard, Zien Alexander. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews] *IEEE Transactions on Neural Networks*. 2009;20:542–542.