Cell instance segmentation using methods from latest publications from CVPR, ICCV, NeurlPS in 2021

1[st] Dias Khalniyasov
Sapienza University of Rome
Rome, Italy
khalniyasov.1954228.@studenti.uniroma1.it

2[rd] Iliyas Bektas
Sapienza University of Rome
Rome, Italy
bektas.1971117@studenti.uniroma1.it

***Abstract*— Instance segmentation is an active research topic in the field of computer vision and deep learning. It is the task of detecting and delineating each distinct object of interest appearing in an image. Despite the existence of a variety of high-accuracy models, in our work we tried to compile different approaches from the best publications of 2021 in this field in order to see how proposed methods will improve the accuracy in comparison to older models. We selected a Sartorius cell instance segmentation dataset from Kaggle. It has 606 labeled images with 73470 annotated objects. In addition, it also has 1972 unlabeled images which can be used for semi-supervised tasks. By using other transfer learning models, adding data augmentation techniques, analyzing different methods, and hyperparameter tuning we were able to increase our intersection over union metric from 8.4% to 24.88%. Unfortunately, we have not been able to beat the proposed baseline of 27.1 %**

***Index Terms*— Computer vision, deep learning, instance segmentation, ResNet, MaskRCNN.**

I. Introduction

Instance segmentation is a challenging computer vision task that requires the prediction of object instances and their per-pixel segmentation mask. This makes it a hybrid of semantic segmentation and object detection. Currently it has become an urgent research topic in the field of computer vision and deep learning with a variety of applications in industries varying from medicine to fashion. Although instance segmentation is well studied research that already has methods with high accuracy, in this project we focus on methods with the best recent scores in this field to see how it is going to improve our score.

In this paper we take the method proposed in [5] as a baseline. In this method, called Mask R-CNN, authors extend Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.. By using other transfer learning models, adding data augmentation techniques, and hyperparameter tuning we were able to increase intersection over union from 8.4% to 24.88%. The rest of the paper is organized as follows. Firstly, in Section2 we review some existing methods. Furthermore, in Section3 we analyze and discuss the chosen dataset. In the next section we describe methodology. In Section4 we discuss experiments and results. Finally, Section5 gives a conclusion of the paper.

II. Dataset

To train our proposed model state-of-the-art dataset we use a dataset from Kaggle [8]. The dataset is a collection of biological images depicting neuronal cell types commonly used in the study of neurological disorders. More specifically, these are microscopy images to train and test models for instance segmentation of neuronal cells. The training annotations are provided as run length encoded masks, and the images are in PNG format. The number of images is small, but the number of annotated objects is quite high. It has 606 labeled images with 73470 annotated objects. In addition, it also has 1972 unlabeled images which can be used for semi-supervised tasks. The hidden test set is roughly 240 image

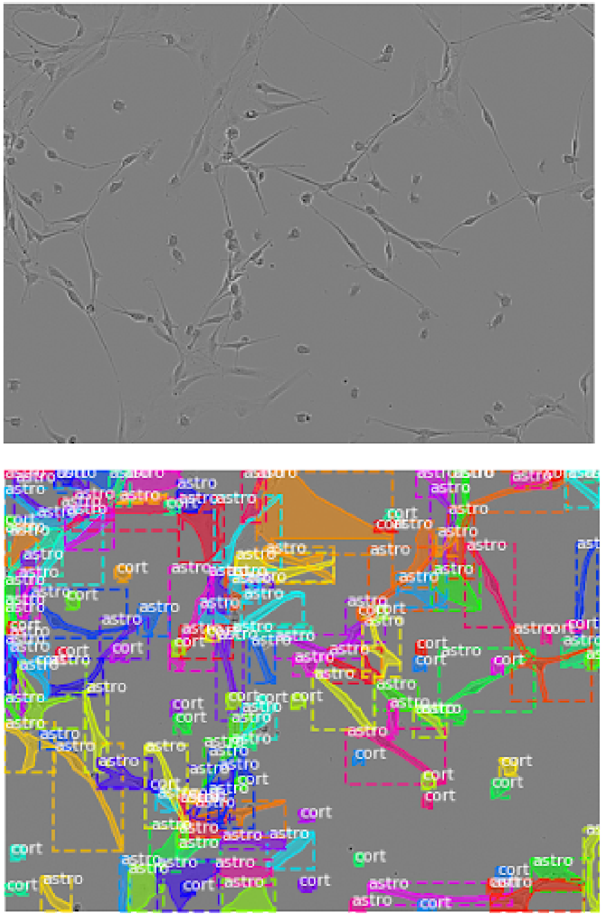
III. Proposed Method Explained

A. Transfer Learning

Firstly, we decided to implement code from papers with architectures. Each member took 2 papers, and tried to tie ready code to a given dataset. The problem was that the given code was click-to-go and not well documented on how to implement it in existing architecture or in what way the dataset needs to be structured. After 4-5 days of unsuccessful tries, we decided to switch to existing notebooks with running code, to experiment with parameters, and reproduce the results. SpineNet [2], DeepMARC [4] were trained on TPU, the experiments with TPU on Kaggle weren't successful, mostly because we have no experience of working with it/debugging. We focused on pre-trained MaskRCNN models using detectron2. As the base code we took [8] as we liked the detailed evaluator and clean code.

B. Data Augmentation

Data augmentation method is used to increase the number of training dataset. We tried to use different combinations of transforms. The key idea behind the Copy-Paste Augmentation Method [1] is to paste objects from one image to another image. We were interested in how augmented data would affect our score. In picture 1.1, you can see an example of how we augmented the data.



Picture 1.1 Original photo(above) and augmented using Copy-Paste(below)

Model name(detectron2)	LR	Augmentation	IoU	Epochs	Total loss:	Train time
mask_rcnn_R_50_FPN_3x	0.0005	Horizontal(0.5), Vertical (0.5)	0.085	10**3	1.962	02:55
mask_rcnn_R_101_FPN_3x	0.0005	Horizontal(0.5), Vertical (0.5)	0.084	10**3	1.962	04:21
mask_rcnn_X_101_32x8d_FPN_3x	0.0005	Horizontal(0.5), Vertical (0.5)	0.0984	10**3	2.009	12:41
mask_rcnn_X_101_32x8d_FPN_3x	0.0001	Horizontal(0.5), Vertical (0.5)	0.2488	10**4		2:25:45
mask_rcnn_X_101_32x8d_FPN_3x	0.0005	Horizontal(0.5), Vertical (0.5)	0.271	10**4	1.08	2:25:45
mask_rcnn_X_101_32x8d_FPN_3x	0.0005	Horizontal(0.5), Vertical (0.5), Copy-Paste, (blend=True, sigma=1, pct_objects_paste=0.5, p=1.0)	0.1753	10**4	1.72	4:59:39

Picture 1.2 Results from training different models

IV. Experiments and Results

The main difficulty here was to not get OOM(Out-Of-Memory) error, so we reduced the number of images per batch (2->1) to be able to fit pretrained models into the GPU.

We learned three available models, and got these results(1.2). Among three tested models we took one with the highest score and trained over 10000 epochs. Reducing initial learning rate(LR=0.0005 ->0.0001) just slowed the learning (we also experimented with LR=0.0003 but stopped the training after a while due to low score in comparison with the best score model). The highest score model still was not able to beat the proposed baseline[9]. Last result represents the model including Copy-Paste augmentation[10]. The result is not promising, we believe mostly because of overlapping objects.

V. Conclusion and Future Work

In conclusion, we gained much experience with preparing data pipelines for training big models (in particular with building custom dataset mappers), transfer learning with limited cloud resources, fine tuning models, and most importantly knowing when to objectively look at the situation and stop doing ineffective methods. Future work could include getting more to know how to work with TPU, adding more geometrical augmentation (Elastic transform, Perspective, GridDistortion and others), implementing Pointly-Supervised Instance Segmentation

REFERENCES

[1] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., & Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr46437.2021.00294

[2] Du, X., Lin, T.-Y., Jin, P., Ghiasi, G., Tan, M., Cui, Y., Le, Q. V., & Song, X. (2020). SpineNet: Learning scale-permuted backbone for recognition and localization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr42600.2020.01161

[3] Dao, P., Lu, J., Li2, H., Mottaghi, R., & Kembhavi, A. (2021). Container: Context Aggregation Network. *2021 Conference on Neural Information Processing Systems*.

[4] Birodkar, V., Lu, Z., Li, S., Rathod, V., & Huang, J. (2021). The surprising impact of mask-head architecture on novel class segmentation. *2021 International Conference on Computer Vision*.

[5] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2017.322

[6] Cheng, B., Parkhi, O., & Kirillov, A. (2021). Pointly-Supervised Instance Segmentation.

[7] Sartorius - cell instance segmentation. Kaggle. (n.d.). Retrieved December 28, 2021, from https://www.kaggle.com/c/sartorius-cell-instance-segmentation/data

[8] Markunys. (2021, December 3). Sartorius transfer learning [train with livecell]. Kaggle. Retrieved December 28, 2021, from https://www.kaggle.com/markunys/sartorius-transfer-learning-train-with-livecell

[9] julian3833. (2021, December 4). Sartorius - Starter Torch Mask R-CNN [lb=0.273]. Kaggle. Retrieved December 28, 2021, from https://www.kaggle.com/julian3833/sartorius-starter-torch-mask-r-cnn-lb-0-273

[10] Lewisgmorris. (2020, April 28). Image segmentation using Detectron2. Kaggle. Retrieved December 28, 2021, from https://www.kaggle.com/lewisgmorris/image-segmentation-using-detectron2