Ukrainian Catholic University

Faculty of Applied Sciences

Sartorius - Cell Instance Segmentation

Author:
Yaroslav Prytula

27 December 2021

Introduction:

The project is based on accurate instance segmentation of the cells. Different neurological disorders, including neurodegenerative diseases such as Alzheimer's and brain tumors, are a leading cause of death and disability across the globe. However, it is hard to quantify how well these deadly disorders respond to treatment. One accepted method is to review neuronal cells via light microscopy, which is both accessible and non-invasive. Unfortunately, segmenting individual neuronal cells in microscopic images can be challenging and time-intensive.
Current solutions have limited accuracy for neuronal cells in particular. In internal studies to develop cell instance segmentation models, the neuroblastoma cell line shsy5y consistently shows the lowest precision scores of eight different cancer cell types tested.
This project proposes the detection that describes distinct objects of interest in biological images representing neuronal cell types commonly used in the study of neurological disorders - shsy5y, astro & cort.

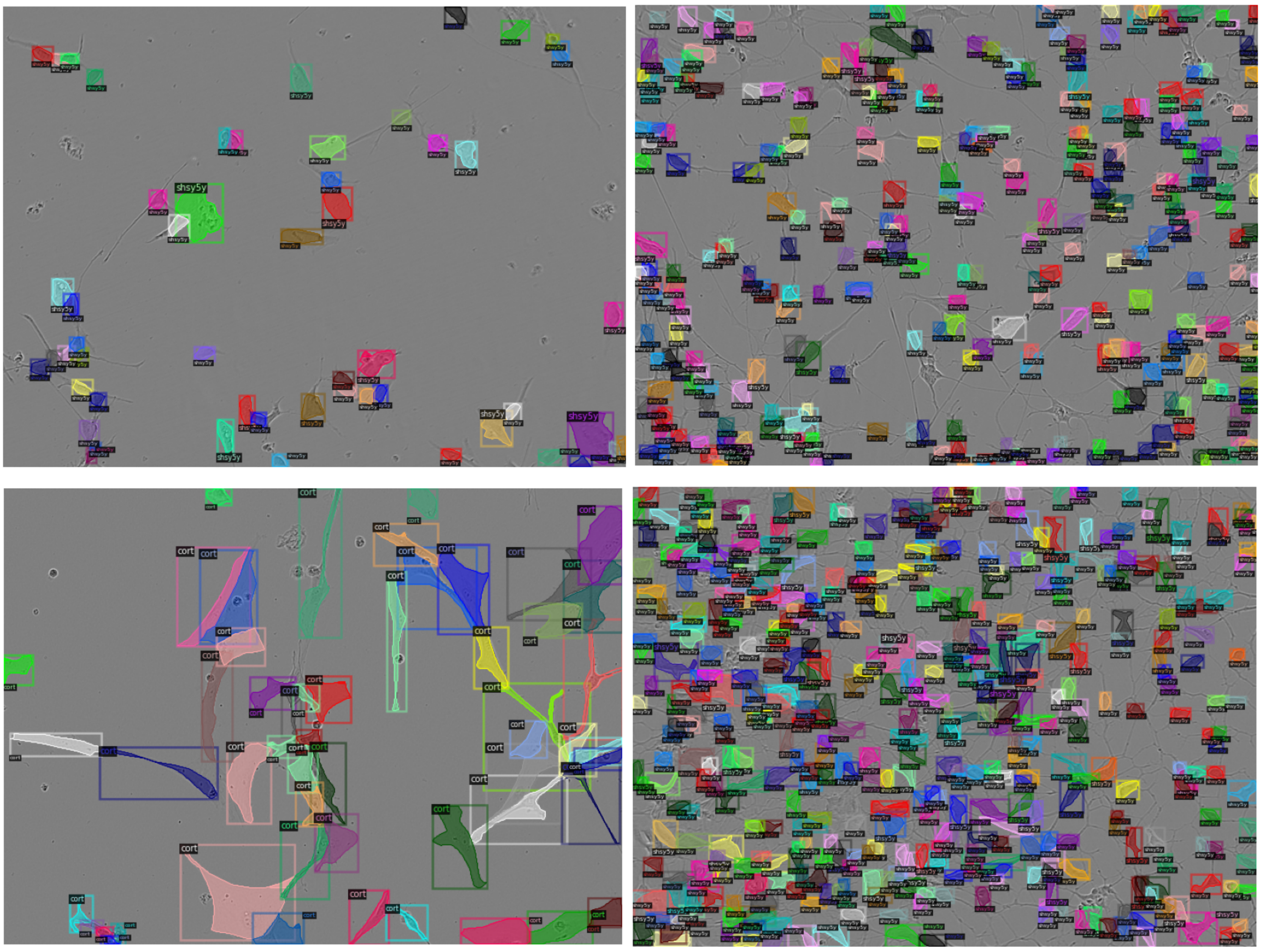
Dataset:

The competition already provided the training annotations 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 relatively high. The hidden test set is roughly 240 images.

The current dataset consists of:

- train - train images in PNG format
- test - test images in PNG format
- train_semi_supervised - unlabeled images offered in case of using the additional data for a semi-supervised approach

Decoded masks represent the instance of different cells. The images below illustrate the sample of data used for training and validation.



Literature Review:

The underlying idea of the project covers different areas of training. Each stage - preprocessing, training, and postprocessing- significantly impacts the score. Indeed, for other models that have been trained on the same data, the output remains somewhat similar. Thus, the postprocessing pipeline is crucial here.
The already existing approaches use some sort of Mask RCNN as the primary source for segmentation. The main problem here is with the mask labeling. Later in this paper, we will find that some mask annotations are broken. The following correlations are also present in the testing dataset. Thus, the approach would be to learn the next error of such masks and to be able to reproduce them.
However, we don't want the error present for accurate natural life cell instance segmentation. Ths the following work will be based on precise cell segmentation. Indeed, even a better scoring model that segments correct labels may produce better scores than the one that knows the error but can't perform well in segmenting the instance of a single cell, especially in large groups.

[1] The approach is suitable for the inference level when we already have a good-performing model and increase its performance.
[2] It is used for building the main pipeline. It uses the Detectron framework to create a pre-trained model and load the coco format dataset.

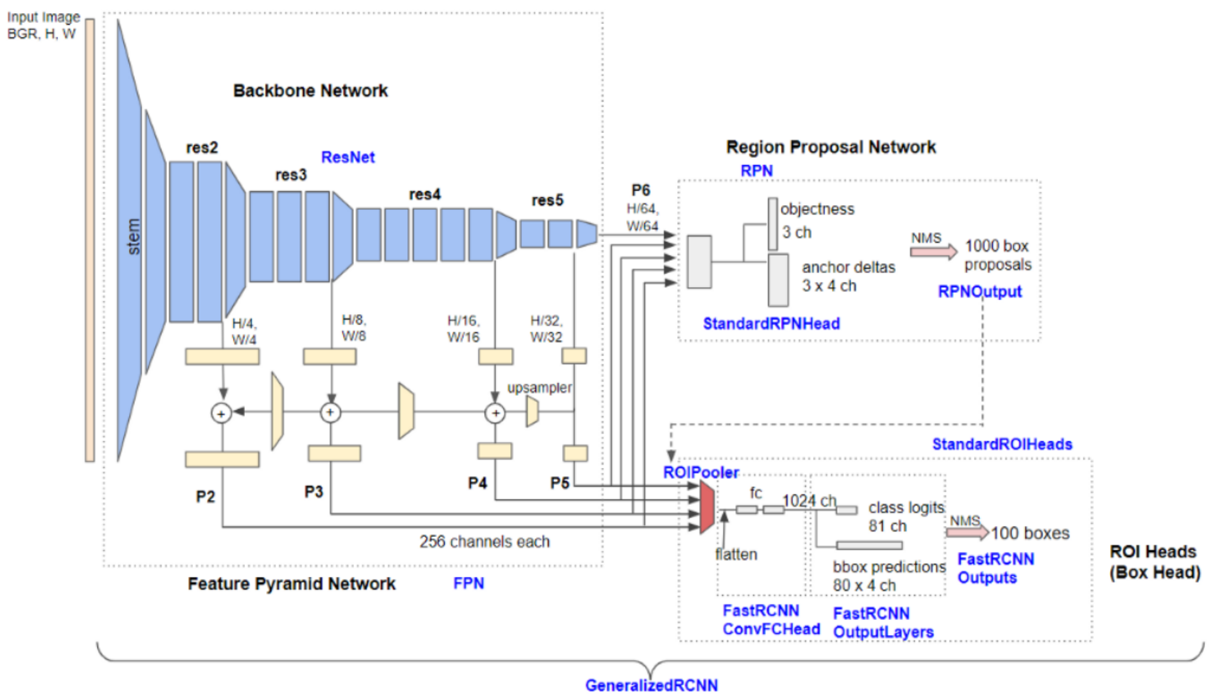
Baseline:

For the baseline, the maskrcnn_resnet50_fpn was used. The overall performance was outstanding, and the model achieved the all-time highest 0.273 AP at box-level IoU. While training, the images were augmented using random rotation and cropping. Such transformation gave the model a little bit of boost. However, this was not enough.

Advanced Pipeline:

- **Detectron2 framework**

  As the final approach model, the Detectron2 framework was used. With Faster R-CNN FPN architecture, the results got significantly better. It uses a multi-scale detector that realizes high accuracy for detecting tiny to large objects. As for the backbone, the model uses the Resnet50. Some experiments were derived with other pre-trained models such as cascade rcnns and deeper results. However, the results were not worth the training time.



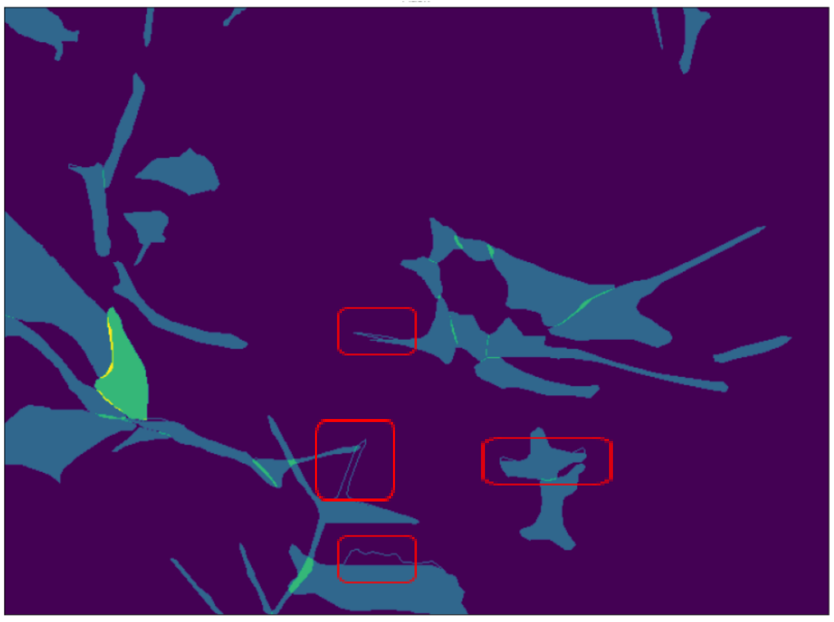
The models have been trained in different conditions. Some were produced by only training on the initial data. Others were trained on the cleaned data. Besides, the multi-scale short size augmentation was applied for the training procedure. However, the models struggled to get any increase from the transforms.

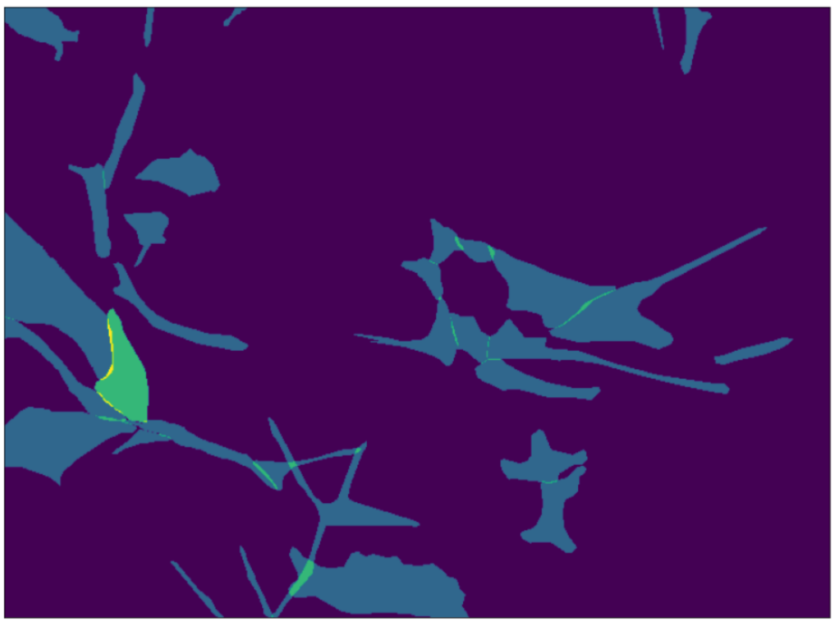
From this point, the [3] paper approach was applied. The model first was pretrained using copy-paste augmentation. Such augmentation uses other images to produce combined cell images with a random position of object segmenting. After, the model was trained with the usual procedure. As a result, the performance level remained the same and no changes were made.

- **Preprocessing**

One of the first bottlenecks present in the given dataset is the broken masks. In [0], the approach is described to fill the masks manually before training and after each image's inference. They are invading the complementary shapes in input from the outer boundary of the image, using binary dilations. Holes are not connected to the edge and are therefore not invaded. The result is the complementary subset of the invaded region. The method used here is based on invading the complementary shapes in input from the outer boundary of the image, using binary dilations. Because the holes are not connected to the border and are therefore not invaded, we can result in the complementary subset of the invaded region.

Here you can observe the problem of unfilled masks and the processed version of them.





- **Postprocessing and Model Ensembling**

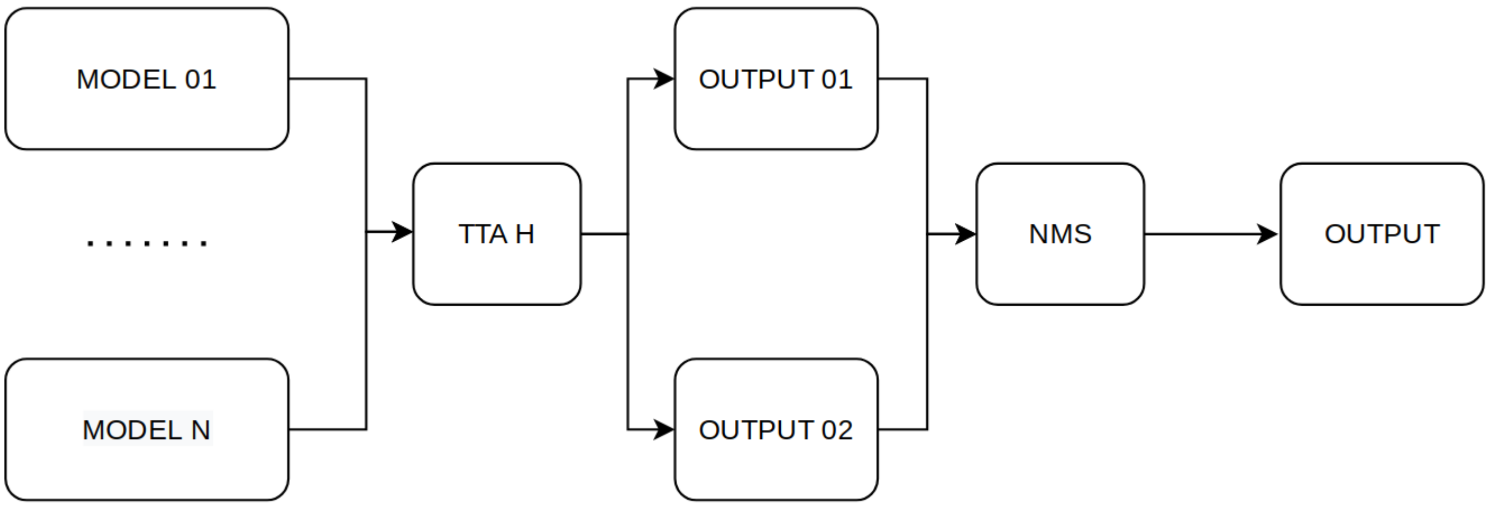
This post-processing part aims to combine the output predicted masks and boxes for cells from different models into one mask that better represents the actual segmentation. Indeed, since different models learn different features especially the models trained in different conditions (with a variety of augmentations), their ensembling can produce better results.

First and foremost, it has been discovered that all three cell classes have different shapes of their initial cells in general. Therefore, the minimum number of pixels for each category was introduced by three thresholdsd. This has drastically changed the result for better. However, for different models, these thresholds are different. Indeed, in almost every situation, the performance of the models

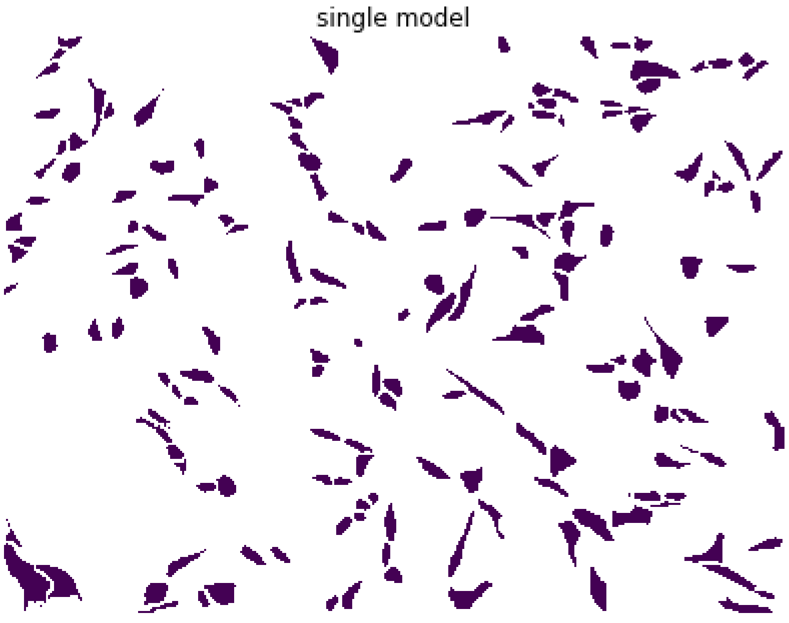
The TTA with horizontal and vertical flips on inference has been used for the baseline approach. However, the performance on the hidden dataset has dropped. After further investigation of that approaches can be used for mask fusion, the pipeline adopted the Non-Maximum Suppression algorithm [4]

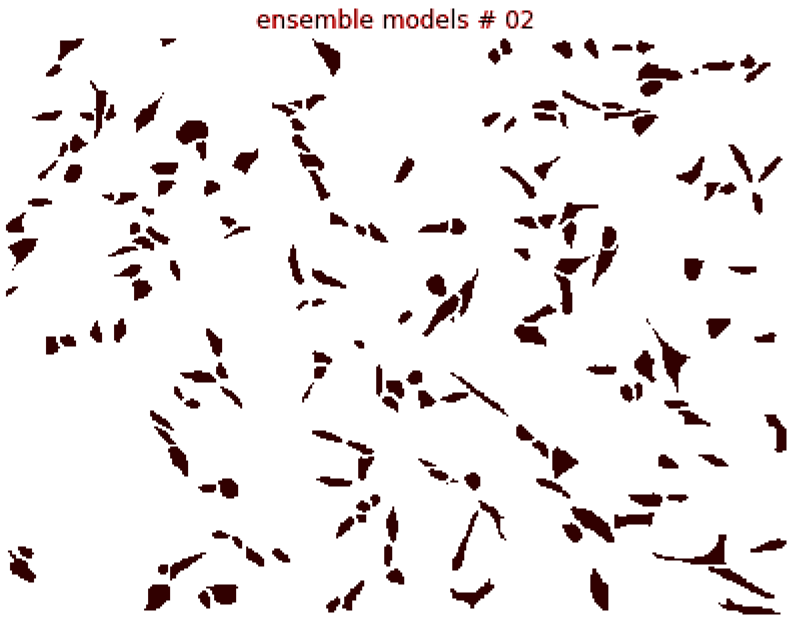
- The format for each output produced by the model is as follows:
  bbox = [x1, y1, x2, y2], class, confidence
- As the first step in NMS, the algorithm sorts the boxes in descending confidences.
- Then, any box that has confidence below this threshold will be removed for some confidence threshold.
- Since the boxes are in descending order of confidence, we know that the first box in the list has the highest confidence. After the first box is removed from the list and add it to a new list
- At this stage, we define an additional threshold for IOU. This threshold is used to remove boxes that have a high overlap. The reasoning behind it is as follows: If two boxes have a significant amount of overlap and belong to the same class, both the boxes are likely covering the same object.
- From this point, the IoU for each box is computed. This procedure is repeated for every box in the image to end up with unique boxes that also have high confidence.

Here you can observe the final for prediction making.



In this part, you can see the segmented cells and their shapes produced by one model. On the other picture is the same situation, but the ensembling flow has been applied


single model


ensemble models # 02

As the result the overall score has increased from 0.307 up to 0.310

Further work:

For the time being the model struggles to learn the segmentation for different cells. Especially, it's hard to do if they are in a large portion of some group. Even after applying the multi-scale short edge augmentation, the model has trouble increasing its performance. This causes many false positives in predictions that lower the score. The reason for this might be the of lack of additional data.
Thus, one planned solution to be used is to create a semi-supervised training pipeline. The dataset, as already mentioned in the very beginning consists of training and validation labeled images. Besides, there are a lot (~2000) unlabeled data photos that can be used. The idea is to have the best ensembling of models to produce the predicted masks for the unlabeled data. After, we can merge the already used and the new data into one large set. Therefore, such an increase in labeled images is expected to create a more robust model to predict on.

References:

[0] https://hal.inria.fr/hal-01757669v2/document
[1] https://towardsdatascience.com/instance-segmentation-automatic-nucleus-detection-a169b3a99477
[2] https://books.google.com.ua/books?id=GfILEAAAQBAJ&pg=PA429&lpg=PA429&dq=using+many+mask+r+cnn+for+mask+better+prediction+pytorch&source=bl&ots=146Kc1rVZD&sig=ACfU3U0Et98OB5QZwCH0AtvsbJugFzeirg&hl=en&sa=X&ved=2ahUKEwifjvifhf7zAhUIGuwKHRxyD3UQ6AF6BAgOEAM#v=onepage&q=using%20many%20mask%20r%20cnn%20for%20mask%20better%20prediction%20pytorch&f=false
[3] https://arxiv.org/pdf/2012.07177.pdf?fbclid=IwAR1rRzRB2TkN-VsHcSKZc67Adaw830Flg2m1BbPBz66_8F35DuEcHJl2G7A
[4] https://arxiv.org/pdf/1411.5309.pdf