Universiteit Leiden

Master Computer Science

Semi-Supervised Iterative Learning on hiPSC-CMs Cardiotoxicity Images

Name:	Adéla Šterberová
Student ID:	s2732815
Date:	18/04/2023
Specialisation:	Bioinformatics
1st supervisor:	Lu Cao
2nd supervisor:	Daan Pelt

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Abstract

Cardiotoxicity can be defined as any damage to the heart or cardiovascular system that emerges from cancer treatment. It is a common side effect of various classes of drugs used to treat cancer. Adverse effects can occur years after the treatment, they can vary from high blood pressure to heart failure, leading to increased mortality among patients. Cardiotoxicity has to be analysed during drug development. For that purpose, human induced pluripotent stem cell-derived cardiomyocytes (hiPSC-CMs) were designed and reprogrammed from human somatic cells. The analysis of microscopy image data acquired from cardiotoxicity studies can be time-consuming for experts because they can obtain a significant amount of image data from high-throughput imaging. Many existing machine learning techniques are capable of identifying objects from images for further analysis, such as Mask R-CNN. However, a considerable amount of labelled data is necessary to train such models. In this paper, we propose a semi-supervised learning method of iterative training. In this method, less annotated data is needed, as the model also trains on annotations predicted from the model trained in previous iterations. The results of the iterative learning method with the use of Mask R-CNN as an inner model had less accurate results in terms of mAP and mF_1 metrics, compared to supervised learning on Mask R-CNN with all data annotated. However, thanks to the new iterative learning method, we found out that Mask R-CNN trained on half of the data annotated and half without annotations can achieve 98% of the precision compared to the model trained on fully annotated data.

1 Introduction

The major adverse effect of anticancer drugs is that they affect the cardiovascular system and may cause damage to the heart, also known as cardiotoxicity. As these effects can vary from myocardial dysfunction to death, it is necessary to observe and evaluate the effects during drug development. Whereas the animal models are not accurate enough for these processes, human induced pluripotent stem cell-derived cardiomyocytes (hiPSC-CMs) [26, 3] can achieve great accuracy for toxicity screening, including all significant genetic variants [9]. Using high-throughput fluorescent microscopy, we can obtain screenings of various conditions of drug-induced toxicity in cells and examine their influence. The possible number of these microscopy images can be high. Furthermore, manual analysis and evaluation of the microscopy image data by experts are typically considerably time-consuming.

There are many different machine learning techniques to detect objects from biomedical image data, mostly convolutional neural networks (CNN), such as Mask R-CNN [16] or U-Net [23]. To be able to conduct further analysis of the cells from the microscopy images, such as comparing the morphological changes and quantifying the measurements, we need to detect each cell instance in the image first. Therefore, we are talking about the task of instance segmentation. Instance segmentation combines the two main computer vision tasks, that is to say, object detection and semantic segmentation. The first task is detecting each respective object and resulting in a bounding box. The second one is a classification of each pixel into a given class [16]. Instance segmentation is a complicated task, especially when the cells under different conditions are clustered together. That decreases the number of models that can be used and increases the number of training data that is needed. The reliable accuracy of the predictions is necessary for the given task, which represents a crucial step for cardiotoxicity analysis.

Getting the data annotations (cell detections in microscopy images) for the training of the models is demanding in terms of time and expense. Therefore, there have been many attempts to improve the learning of the model on a smaller amount of labelled data. We propose a semi-supervised learning model that is based on the work of [7]. In this model, only a smaller amount of labelled data is needed at the beginning, and the model uses its own predicted labels for further training of the model.

In this paper, we introduce the iterative learning model incorporating Mask R-CNN network architecture for instance segmentation. We train and test the proposed model on the data from experiments imitating cardiotoxicity on hiPSC-CMs obtained from fluorescence high-throughput imaging analysis [9]. We evaluate the proposed model, and subsequently, we discuss the influence of the *score* predicted for each mask as the label for further training. We compare the model with the supervised model trained on fully labelled data. Lastly, we describe limitations and possible improvements.

1.1 Biological Background

Chemotherapy is an anticancer therapy with the use of anticancer drugs that has great potential for healing the patient. There are many different kinds of drugs that target specific molecules in the organism to prevent the spread of cancer. Even though there has been progress in the development of these drugs in the past years and the survival rate of the patients has increased, a common issue appears - cardiotoxic side effect [31]. It applies to a number of anticancer drugs from different classes, such as doxorubicin [4], crizotinib [27], or sunitinib [30]. Hence, cardiotoxicity needs to be tested and evaluated during the drug development process.

As the animal models are not sufficient for modelling the cardiotoxic effects [29], a system of reprogramming human somatic cells into induced pluripotent stem cells (hiPSC) [26] was developed in 2007. In the following years, the development of differentiation of hiPSC into different cell types improved the efficiency of these in vitro systems [3]. Additionally, the cardiomyocytes derived from hiPSC (hiPSC-CMs) enhanced the screening and evaluation of cardiotoxicity with the possibility to include different genetic variants in the population during the drug development [8, 25].

There is a variety of technologies available for the evaluation of experiments with cardiotoxicity on hiPSC-CMs and detecting bio-moleculs. The most commonly used technology is fluorescence readout that reflects electrical and calcium signals over time [5]. Another one is high-throughput image analysis, which is an automated light microscopy and image analysis. It can identify changes in α-actinin signal in hiPSC-CMs [12].

1.2 Overview

The paper is organised as follows. Firstly, the section 2 describes the data, and how they were obtained and preprocessed. It introduces the models that were used and how they work, and the metrics used to evaluate them. Next, the section 3 describes algorithms, implementation choices, and experimental setup. Following, the section 4 introduces and interprets the results from the experiments. Lastly, in the section 5, the knowledge from the previous chapters is summarized, and possible improvements are suggested.

2 Background and Data

In this section, we will describe the background of models used for the training of instance segmentation tasks, how the data that we worked with were obtained and what metrics to evaluate the performance were used.

2.1 Data

The data for this purpose were obtained by treating the hiPSC-CMs with different concentrations of anticancer drugs. The density of the cells was 10 000 cells per well on Corning 96-well cell culture plates. The cells were then treated with two anticancer drugs - doxorubicin and crizotinib [9]. The dosing range was from $0.1\mu M$ to $10\mu M$. It resulted in 5 experiments captured under 12 different conditions. Subsequently, the antibody was used on α-actinin binding protein to identify the structure of sarcomeres inside cardiomyocytes. To detect the nuclei in cells, they were stained with blue-fluorescent DNA substance - DAPI [9].

The images were captured by high-throughput BD pathway 855 microscopes. The microscopes were equipped with an Olympus 20xLWD objective (NA 0.75) and a Hamamatsu ORCA-AG CCD digital microscope camera. The first signal of nuclei stained with DAPI was obtained using a $380/10 - nm$ excitation filter with $0.0078s$ exposure time and the $435LP$ Chroma emission filter. The second signal of α-actinin cells was obtained using a $HQ548/20$ excitation filter with $0.08s$ exposure time plus 2 gains and the $84101m$ Chroma emission filter. The cell cultures were scanned by a 7x7 montage setup resulting in images of size 4700x3600 pixels [9]. The example of gained images can be seen in Figure 1.

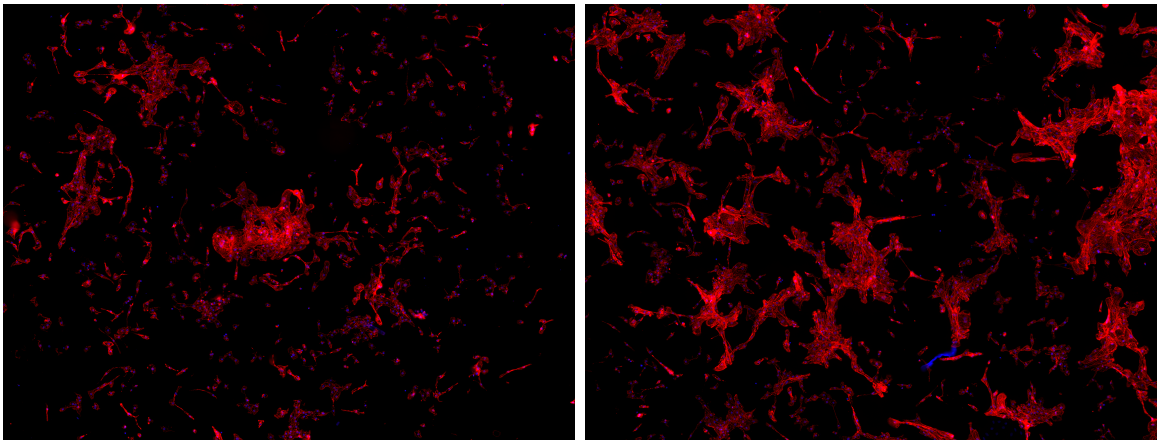


Figure 1: Examples of images from different conditions acquired by BD pathway 855 microscope, 7x7 montage setup [9]. The red channel shows the α-actinin signal with cell structures. The blue channel shows the DAPI signal with stained nuclei.

As we were not able to have data annotated by experts, we use the results from the image analysis system proposed by [9], as ground truths for our model (labels). The resulting masks achieve a precision of 0.83 and a recall of 0.93, compared to the original manually segmented images. The segmented results come from an automated pipeline designed in ImageJ software [2], as seen in Figure 2. It consists of preprocessing, such as background subtraction, Gaussian smoothing filter or contrast enhancement. Then it is divided into two parts. The first is nuclei detection - by using watershed segmentation. The second is cell mask detection - mainly using the Otsu thresholding

method. Then those two outputs are combined in the next step of seeded propagation - using the nuclear propagation approach. It means using the nuclei mask as a seed and then propagating in the mask of all cells (foreground) to determine the border of each specific cell. This method resulted in instances of cell masks [9].

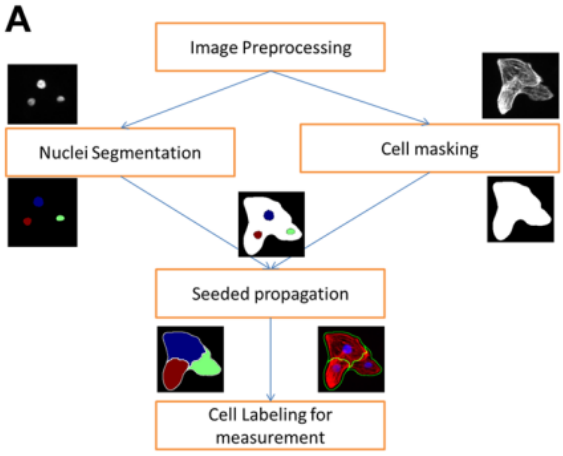


Figure 2: A pipeline for automated image analysis for instance segmentation of individual cells [9].

2.2 Mask R-CNN

Mask R-CNN is the most commonly used method for instance segmentation in the computer vision domain. Its implementation is rather simple, but very robust for all kinds of tasks, and it outperforms most of the state-of-the-art methods. Specifically, it claims to outperform all winners of the COCO challenge [20].

The implementation of Mask R-CNN is based on the architecture of the Faster R-CNN [22]. Furthermore, the Faster R-CNN method is extended, more rapid adaptation of the Fast R-CNN architecture [13], according to speed-accuracy study [17]. R-CNN denotes Region-based CNN, which combines region proposals with standard CNN [14].

The architecture of Faster R-CNN consists of two main parts. The first one is Region Proposal Network (RPN), which is an attention mechanism proposing the possible candidate regions - resulting in bounding boxes [22]. As a backbone network, different network models, such as ResNet, can be used. The second part comes from Fast R-CNN architecture, where the goal is to extract features from each Region of Interest (RoI) by RoIPool operation. Finally, it performs the regression on bounding boxes and the classification of objects. Therefore, the two outputs from Faster R-CNN are a bounding box and a classification label for each candidate object [13].

Mask R-CNN is a simple add-on branch to the Faster R-CNN architecture that predicts the binary mask from the given RoI. Hence, it adds a third output in the form of a mask to the bounding box and the class from the default architecture. For the Mask

R-CNN training, the multi-task loss function is defined by

$$L = L_{cls} + L_{box} + L_{mask}, \tag{1}$$

where L_{cls} is classification loss and L_{box} is loss of bounding box defined in [13]. L_{mask} is defined as the average binary cross-entropy loss for the k-th mask on RoI of ground truth class k [16].

A simplified scheme of the Mask R-CNN framework can be seen in Figure 3. The crucial part is the newly proposed RoIAlign layer, which preserves and connects the extracted features with the given input. It is a quantization-free layer that impacts mask accuracy, as the quantization in the previous RoIPool layer would cause misalignment between the feature map and RoI [16]. The RoIAlign layer removes the quantization using bi-linear interpolation and aligns the features with the inputs [19].

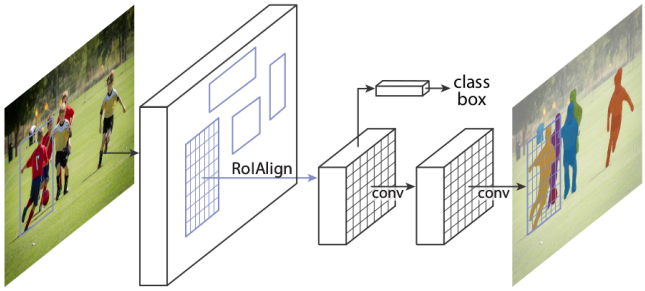


Figure 3: A simplified visualization Mask R-CNN architecture for instance segmentation that extends the Faster R-CNN model [16].

2.3 Semi-Supervised Learning Using Sparse Labelling

Semi-supervised learning is an increasingly used method with the ability to maintain model accuracy using fewer data. Hence, it saves time for data annotation without a significant loss of accuracy and enhances time efficiency. Semi-supervised learning using sparse labelling was proposed in [11], where U-Nets were proved to be valid models for such training. In the paper [7], they enhanced the model with an iterative training approach combined with sparse labelling, which yields an accuracy of 90% compared to training on fully annotated data.

The core model used for this method is U-Net, slightly modified for the purpose of microscopy images [24]. U-Net, in general, is a CNN designed for the segmentation of biomedical images that can be trained on a small number of samples [23]. In this implementation, it is a 3-block architecture with a 32-size filter in the first block, where encoder-decoder feature maps are concatenated [24].

A scheme of the method of iterative learning can be seen in Figure 4. It shows how the iterative training of the U-Net works. Firstly, the sparsely annotated dataset is created by combining labelled and unlabelled data or extracting the labels from the

annotated data. Then the U-Net is trained on this dataset, and the model is used to predict labels from unlabelled data [7]. The next step is the postprocessing of the detected labels. In [7], the detected labels are filtered according to the mean confidence of each object, which must be higher than 0.8. For smoothing the boundaries, the morphological closing with a 3x3 structuring element is applied. Then the unlabelled data are overwritten with the postprocessed labels predicted from the previous model and combined with originally labelled data for the next iteration of the training. On this new dataset, an advanced model can be trained with U-Net, etc. One of the steps of the method includes random augmentation to gain random noise and scale so that the model is able to detect slightly different objects than the original labels [28].

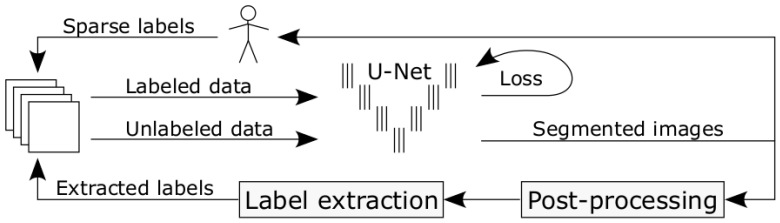


Figure 4: A visualization of the iterative learning. The training starts with sparse labels, and then the new labels are predicted from the trained model and included in the next iteration of training in the loop [7].

2.4 Metrics

To evaluate and compare the models, we need to define the metrics used to calculate the performance of the models. Firstly, we need to introduce an intersection over union (IoU) score. It is a fraction between an intersection and union of the predicted mask M_P and ground truth (GT) mask M_{GT} given by

$$IoU = \frac{area(M_P \cap M_{GT})}{area(M_P \cup M_{GT})}. \tag{2}$$

With IoU defined, we can also distinguish correct and incorrect predictions. Masks that were predicted correctly are true positive (TP), which means that their IoU value is higher than certain threshold t, $IoU \geq t$. On the other hand, if $IoU < t$, the mask is labelled as a false positive (FP). Also, if there are more predicted masks for one GT mask, only one of them is considered TP and the remaining ones are FP. The number of ground truth masks that were not predicted is equal to the number of false negative (FN) masks [15].

Precision is a proportion between correctly labelled masks and all predicted masks given by

$$precision = \frac{TP}{TP + FP}, \tag{3}$$

where TP+FP is the total number of all predicted masks. Recall is a proportion of

the number of correctly labelled masks to all relevant masks given by

$$recall = \frac{TP}{TP + FN}, \tag{4}$$

where shortcuts TP+FN is the total number of ground truth masks. The F_1 score is a harmonic mean between precision and recall [21] and is defined as

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{5}$$

However, according to paper [13], the F_1 score might not be accurate enough as the metrics for an instance segmentation and a mean average precision metric (mAP) is proposed. It is defined by precision-recall relation. The precision is interpolated $p_{interpolation}$ at all unique recall values spaced between 0 and 1. Where at each recall value r, $p_{interpolation}$ is the highest value found for any recall value higher than r. It is given by the equation

$$p_{interpolation}(r) = \max_{r' \geq r} p(r'). \tag{6}$$

Then we can define average precision (AP) as the area under an interpolated precision curve given by

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interpolation}(r_{i+1}), \tag{7}$$

where r_i is the i-th recall value, and n is the number of values. Then the mAP is the mean of average precision for each class, which is only one in our case.

3 Methods

We decided to compare the two methods - fully supervised learning with our own implementation of semi-supervised learning. In this section, we describe implementation details, such as the algorithm, parameter choices or experimental setup.

To create a dataset for training the models, we cropped all images into square images of size 512x512 pixels. Each microscopy image had 3 channels - red, green, and blue. However, the green channel was set to zero in all images. Because we only worked with the red and blue channels, depicting α-actinin and DAPI signal, respectively. Hence, the dimension of the image was 512x512x3. Each microscopy image was complemented by images of masks. Each mask bounded to the microscopy image was documented separately in the form of a binary image, where only the foreground (mask) and background of the image were distinguished. Therefore, the number of images with masks for each microscopy image was the same as the number of cells in the image. Subsequently, 1000 samples of these images were randomly chosen for training, 100 samples for validation and 100 samples for testing the models.

3.1 Supervised Learning

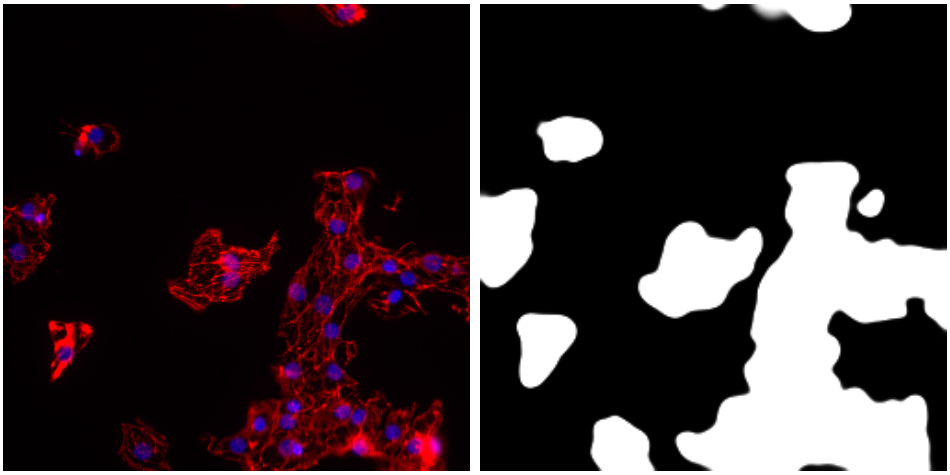
Supervised learning experiments were done with Mask R-CNN implementation based on [1], which is an open-source project under the MIT licence, and it is based on the deep learning framework Keras [10]. The details of the architecture of the Mask R-CNN model can be found in the subsection 2.2. Each training of the Mask R-CNN model used pre-trained COCO weights trained on data from [20]. It is a common practice of transfer learning used on complex models, where the training does not need to start from scratch but has already encountered some real-world data.

For the Mask R-CNN implementatisuccessfully on, hyper-parameter tuning is a very difficult task due to the computational complexity and the duration of the training. Therefore, most of the hyper-parameters were used as suggested in [16], as they are claimed to be robust enough for most of the tasks. Hence, only a few parameters were examined. A simple grid search was done to find suitable parameters for learning with respect to the time complexity. We examined the number of epochs from the interval [10, 60], the number of images per GPU from the interval [1,8], and the learning rate from values [0.0001, 0.001, 0.01]. The best performance was achieved with the number of epochs set to 50, 6 images per GPU and the learning rate l_r set to 0.001. Regarding the number of data samples for training and validation and the number of images per GPU, the number of steps per epoch and validation steps were set to 166 (1000/6) and 16 (100/6), respectively. It is the number of samples available for each process divided by the number of images per GPU. As a backbone network, we used ResNet50 with a depth of 50 layers. The mask was considered TP when an IoU score reached a higher value than 0.5.

3.2 Semi-Supervised Learning

Semi-supervised method using iterative learning by adding newly predicted labels to the ground truth labels using U-Net as the inner model represents a promising method for the data that are difficult to annotate [7]. Even though U-Net CNN is a valid model for biomedical images and can be trained from only a few samples of data [23], it is not applicable to our dataset. The U-Net CNN is a model for semantic segmentation. Furthermore, the cells in our image data are highly clustered. Therefore, the U-Net CNN is not a suitable model for our task of instance segmentation. It can be noticed in Figure 5, where we can see that the U-Net distinguished the background and foreground but failed to label each instance of the cell separately.

Based on [7], we propose an iterative learning model using the Mask R-CNN [16] as the inner model. The scheme of this training can be seen in Figure 6. We begin the training of the model with half of the data (500 samples) randomly chosen to keep the mask labels, and for the rest of the data (the other 500 samples) the annotations are removed. After each training iteration, the data that are not labelled are used to create a prediction of new masks by the trained model. The predictions are then post-processed and filtered according to their *score*. Each cell mask is filtered separately.



(a) An input image for the U-Net (b) Predicted mask

Figure 5: An example of usage of the U-Net model to predict masks on a sample image from our dataset cropped to 512x512 pixels.

We experiment with three different thresholds for this *score*. We start at a *score* of 0.8, which is suggested in [7]. However, it is a slightly different score, coming from a different method, but with the same meaning - confidence in the prediction. Moreover, we are increasing the threshold to 0.9 and 1.0. Chosen masks are then added to the originally labelled data for the training of a new model, same as in [7].

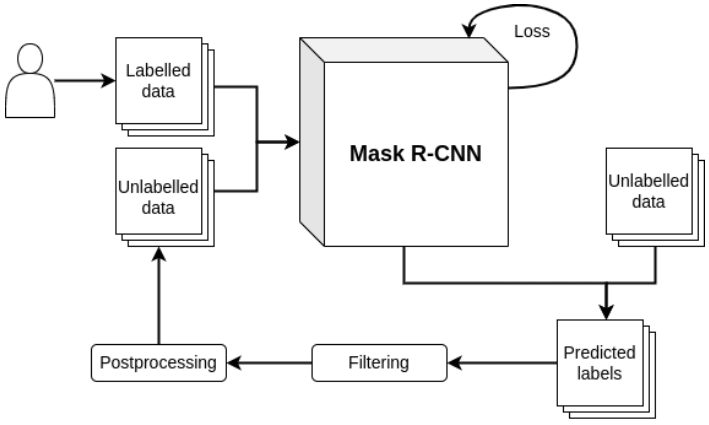


Figure 6: A visualization of the iterative learning incorporating Mask R-CNN as an inner backbone model based on the iterative learning method proposed in [7]. The Mask R-CNN is trained at each iteration on the labelled data, and the unlabelled data or data with predicted labels.

Postprocessing techniques are applied to all predicted masks from the Mask R-CNN model as well as those from the iterative learning model. Postprocessing is applied on each mask separately so that no mistakes are done on predictions, e.g. connecting two different masks by postprocessing method. We apply the morphological closing with structuring element 3x3 to smooth the boundaries as in [7]. Afterwards, we fill

the holes in predictions, as we know the masks should be in one piece with integrity intact. Lastly, we filter the masks that are too big, to be real. For example, masks that cover most of the image are filtered, because they are mistakes made by faulty data or model errors.

3.3 Experimental Setup

All experiments were done with three repetitions. They were running on 2 NVIDIA GeForce RTX 2080 Ti with 16 Intel Xeon E5-260 cores and a processor base frequency of 2.00 GHz. Most of the parameters for the Mask R-CNN were set to values suggested in [16]. Some of the most important ones can be found in Appendix A.

4 Results

In this section, we provide results from experiments on the supervised model - Mask R-CNN and semi-supervised model - iterative learning.

4.1 Supervised Learning Results

Results in Table 1 mainly serve as a benchmark to compare the results of semi-supervised iterative learning. The score of mAP and mF_1 are relatively high, and the results seem to be stable because the standard deviation is very low. The number of predicted masks from the test set was, on average, 929, of which 904 predicted masks have a high certainty. The test set consisted of 1075 masks in total, which means that the model predicted around 86% of the masks, from which 97% were of high confidence.

Table 1: Results from the supervised learning experiments. We note the mean and standard deviation for each metric. n means the number of cell masks predicted by the model, and ≥ 0.8 denotes how many of these masks have scored higher than threshold 0.8.

metrics	$\boldsymbol{mAP}$	$\boldsymbol{mF_1}$	$\boldsymbol{n}$	$\boldsymbol{\geq 0.8}$
mean	0.7477	0.5935	929	904
std	0.0155	0.0069	29	30

4.2 Semi-Supervised Learning Results

Results of mAP and mF_1 scores from semi-supervised learning experiments can be found in Table 2. It shows results from four iterations of the iterative learning model.

We experimented with three different thresholds of the *score* $(0.8, 0.9, 1.0)$, according to which the predicted masks that should be involved for the next iteration of training the model are filtered. Results from the same experiments counting the number of predicted cells in general or above the threshold are noted in Table 3.

In Table 2, we can observe that the highest mAP is in the first iteration in all three experiments. In the case of the *score* of 0.8, it decreases rapidly with each iteration. For the *score* of 0.9, it also decreases but less rapidly. In the case of the *score* of 1.0, the mAP score has some increasing tendencies between the second and third iterations. However, it is still much lower in iterations II, III, and IV than in iteration I. In all three experiments, the mAP score decreased from 0.73 in the first iteration I to around 0.65-0.67 in iteration IV.

The same can be observed when looking at the mF_1 score. With one exception in iteration III and the experiment with the *score* of 1.0, the model reached the highest value of mF_1 0.65, which exceeded the mF_1 score from the first iteration I and even the score of mF_1 from supervised learning (where it reached 0.59). However, the value 0.65 has a standard deviation of 0.08, which is a very high value. Hence, we can conclude that the training of this model was unstable, and repetitions differed significantly. Except for that, the mF_1 score decreased with each iteration resulting in the lowest values in iteration IV for all scores.

The value of the number of cell masks predicted fluctuated in all iterations and all experiments, which can be seen in Table 3. On average, the number of predicted cells was the highest in the experiment where we used the *score* of 0.9. It was, on average, 943 cell masks. The test dataset obtained 1075 masks in total. Hence, it predicted around 87% masks regardless of the accuracy.

4.3 Prediction Examples

The examples of model predictions can be found in Appendix B. We chose two samples of data with a high density of cells and cells clustered together, which are challenging cases for the model to predict the mask instances properly. Figure 7 - Figure 11 show the results from the best trained models. In Figure 7, we can see examples of prediction from supervised learning, where Mask R-CNN was trained on all annotated data. Figure 8 shows examples of prediction from a model trained on half-annotated/half-unannotated data, which is the first iteration of iterative learning. Figure 9 - Figure 11 show the examples of predictions from models of iterative learning, where different values of the *score* were used to filter the masks - 0.8, 0.9 and 1.0. All predictions come from the model trained in iteration IV.

Table 2: Results of mAP and mF_1 scores from semi-supervised learning experiments. It notes four iterations of iterative learning for three different thresholds $(0.8, 0.9, 1.0)$ of *score* to filter the masks for the next training. The means and standard deviations for each metric are noted.

		iter I		**iter II**		**iter III**		**iter IV**	
score		$\boldsymbol{mAP}$	$\boldsymbol{mF_1}$	$\boldsymbol{mAP}$	$\boldsymbol{mF_1}$	$\boldsymbol{mAP}$	$\boldsymbol{mF_1}$	$\boldsymbol{mAP}$	$\boldsymbol{mF_1}$
0.8	mean	0.7336	0.5900	0.7157	0.5789	0.6575	0.5534	0.6563	0.5549
	std	0.0112	0.0083	0.0265	0.0137	0.0466	0.0210	0.0165	0.0086
0.9	mean	0.7336	0.5900	0.7171	0.5815	0.6893	0.5711	0.6792	0.5605
	std	0.0112	0.0083	0.0130	0.0060	0.0315	0.0179	0.0342	0.0177
1.0	mean	0.7336	0.5900	0.6261	0.5402	0.6798	0.6467	0.6546	0.5508
	std	0.0112	0.0083	0.0419	0.0177	0.0268	0.0820	0.0274	0.0166

Table 3: Results of a number of cells n from semi-supervised learning experiments. It notes four iterations of iterative learning for three different thresholds $(0.8, 0.9, 1.0)$ of *score* to filter the masks for the next training. The means and standard deviations for each metric are noted. n means the number of cell masks predicted by the model, and ≥ 0.8 denotes how many of these masks have scored higher than threshold 0.8.

		iter I		**iter II**		**iter III**		**iter IV**	
score		$\boldsymbol{n}$	$\boldsymbol{\geq 0.8}$	$\boldsymbol{n}$	$\boldsymbol{\geq 0.8}$	$\boldsymbol{n}$	$\boldsymbol{\geq 0.8}$	$\boldsymbol{n}$	$\boldsymbol{\geq 0.8}$
0.8	mean	946	918	956	932	901	874	901	875
	std	15	6	31	33	52	58	18	17
0.9	mean	946	918	946	923	943	916	936	908
	std	15	6	18	30	61	59	21	14
1.0	mean	946	918	732	692	848	814	815	782
	std	15	6	118	124	88	100	86	79

5 Discusssion

It could not be proven that the iterative learning model with added predicted labels can have comparable precision to the Mask R-CNN model trained on labelled data, as the model was not improving with more iterations of learning. On the other hand, we can conclude that the model trained on half of the labelled and half of the unlabelled data (the first iteration of iterative learning) performed almost as well as the model trained on the data where all labels were assigned. The iterative learning model (iteration I) achieved around 98% of mAP and around 99% of mF_1 score compared to the supervised learning model.

The possible reason why iterative learning (with adding new predicted annotations among the data to train the model) worked with the U-Net model and not with the Mask R-CNN model, is that the performance comparable to the fully supervised training was already proved with a fraction of sparse or dense annotated data used to train the U-Net in [6]. However, we could not find a similar study for the Mask R-CNN model. Also, semantic segmentation tasks done by U-Net differ greatly from instance segmentation tasks performed by the Mask R-CNN model.

Another possible reason is that the predicted mask *score* from the Mask R-CNN model is not measured accurately. It is vaguely defined as confidence score [16], and it is returned from the Keras framework of neural networks [10]. Moreover, the *score* does not seem to be correlated to the IoU score metrics according to [18]. The predicted *score* is quite important for the iterative learning model. Based on the *score* value, the predicted masks are included in or excluded from the data for the next training.

This is the reason why a new scoring mechanism is proposed to return more adequate results in [18]. Moreover, the *score* values aligned to each predicted cell mask are usually very high, as can be seen in Table 1 and Table 3. The total number of predicted cell masks is very close to the number of masks with the $score \geq 0.8$. It might also be given by the fact that only one class can be predicted in our data, which is the cell. It allows the Mask R-CNN model to be quite confident about its predictions. Therefore, the model returns high scores of masks. However, by adding the framework of better predictions of confidence score using [18], we could only include masks with a more precisely determined value of confidence for the next iteration of training. Hence, training the model on better masks could yield better results.

6 Conclusion

In this paper, we proposed the semi-supervised iterative learning model. The model contained the Mask R-CNN model as a backbone network architecture that is appropriate for instance segmentation tasks. During the iterative learning, the model used predicted labels from the previous iteration to improve the precision of the model. We compared the proposed semi-supervised model to the fully supervised model, which

was Mask R-CNN trained on all labelled data.

We proved that training a semi-supervised model with half of the data annotated and half without annotations (meaning only the first iteration of iterative learning) has comparable results to a supervised learning model trained on fully annotated data, using Mask R-CNN as the model. More precisely, the model achieved around 98% of mAP compared to the supervised learning model trained on all labels.

The study can be extended with other ratios of the amount of annotated data and how it affects the results compared to fully annotated data. Moreover, it can define the ratio of annotated/unannotated data that still has profitable results without losing accuracy. It could save the time of the experts for unnecessary annotations.

Iterative learning model [7] combined with the Mask R-CNN framework [16] could not yield comparable results to the supervised learning with only annotated data. However, a possible improvement for the iterative learning model could be combining the Mask R-CNN architecture with the Mask scoring R-CNN [18] framework. Such a combination has a higher chance of delivering better results in future work.

Acknowledgments.

I would like to acknowledge Lu Cao for creating and providing the data from the cardiotoxicity experiment and guiding me through them. Furthermore, I would like to thank her for her time and endless discussions about the project and possible directions.

References

[1] Waleed Abdulla. Mask R-CNN for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[2] Dr. Michael D. Abràmoff, Dr. Paulo J. Magalhães, and Dr. Sunanda J. Ram. Image processing with imagej. *Biophotonics International*, 11(7):36–42, July 2004.

[3] B D Anson, K L Kolaja, and T J Kamp. Opportunities for use of human iPS cells in predictive toxicology. *Clin Pharmacol Ther*, 89(5):754–758, March 2011.

[4] Gregory T. Bass, Karen A. Ryall, Ashwin Katikapalli, Brooks E. Taylor, Stephen T. Dang, Scott T. Acton, and Jeffrey J. Saucerman. Automated image analysis identifies signaling pathways regulating distinct signatures of cardiac myocyte hypertrophy. *Journal of Molecular and Cellular Cardiology*, 52(5):923–930, May 2012.

[5] Stephane Bedut, Christine Seminatore-Nole, Veronique Lamamy, Sarah Caignard, Jean A Boutin, Olivier Nosjean, Jean-Philippe Stephan, and Francis Coge. High-throughput drug profiling with voltage- and calcium-sensitive fluorescent probes in human iPSC-derived cardiomyocytes. *Am J Physiol Heart Circ Physiol*, 311(1):H44–53, May 2016.

[6] John-Melle Bokhorst, Hans Pinckaers, Peter van Zwam, Iris Nagtegaal, Jeroen van der Laak, and Francesco Ciompi. Learning from sparsely annotated data for semantic segmentation in histopathology images. In M. Jorge Cardoso, Aasa Feragen, Ben Glocker, Ender Konukoglu, Ipek Oguz, Gozde Unal, and Tom Vercauteren, editors, *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*, volume 102 of *Proceedings of Machine Learning Research*, pages 84–91. PMLR, 08–10 Jul 2019.

[7] Roman Bruch, Rüdiger Rudolf, Ralf Mikut, and Markus Reischl. Evaluation of semi-supervised learning using sparse labeling to segment cell nuclei. *Current Directions in Biomedical Engineering*, 6(3):398–401, September 2020.

[8] Paul W Burridge, Yong Fuga Li, Elena Matsa, Haodi Wu, Sang-Ging Ong, Arun Sharma, Alexandra Holmström, Alex C Chang, Michael J Coronado, Antje D Ebert, Joshua W Knowles, Melinda L Telli, Ronald M Witteles, Helen M Blau, Daniel Bernstein, Russ B Altman, and Joseph C Wu. Human induced pluripotent stem cell-derived cardiomyocytes recapitulate the predilection of breast cancer patients to doxorubicin-induced cardiotoxicity. *Nat Med*, 22(5):547–556, April 2016.

[9] Lu Cao, Andries D. van der Meer, Fons J. Verbeek, and Robert Passier. Automated image analysis system for studying cardiotoxicity in human pluripotent stem cell-derived cardiomyocytes. *BMC Bioinformatics*, 21(1):187, May 2020.

[10] François Chollet et al. Keras. https://keras.io, 2015.

[11] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing.

[12] Kimberly R Doherty, Robert L Wappel, Dominique R Talbert, Patricia B Trusk, Diarmuid M Moran, James W Kramer, Arthur M Brown, Scott A Shell, and Sarah Bacus. Multi-parameter in vitro toxicity testing of crizotinib, sunitinib, erlotinib, and nilotinib in human cardiomyocytes. *Toxicol Appl Pharmacol*, 272(1):245–255, May 2013.

[13] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[14] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 11 2013.

[15] Allen Goodman et al. 2018 data science bowl. https://kaggle.com/competitions/data-science-bowl-2018, 2018.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[17] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[18] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. *CoRR*, abs/1903.00241, 2019.

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[21] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *CoRR*, abs/2010.16061, 2020.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[24] Tim Scherr, Andreas Bartschat, Markus Reischl, Johannes Stegmaier, and Ralf Mikut. Best practices in deep learning-based segmentation of microscopy images. In *Proceedings - 28. Workshop Computational Intelligence, Dortmund, 29. - 30. November 2018. Ed.: F. Hoffmann*, pages 175 – 195. KIT Scientific Publishing, 2018. 47.01.02; LK 01.

[25] Daniel Sinnecker, Karl-Ludwig Laugwitz, and Alessandra Moretti. Induced pluripotent stem cell-derived cardiomyocytes for drug development and toxicity testing. *Pharmacol Ther*, 143(2):246–252, March 2014.

[26] Kazutoshi Takahashi, Koji Tanabe, Mari Ohnuki, Megumi Narita, Tomoko Ichisaka, Kiichiro Tomoda, and Shinya Yamanaka. Induction of pluripotent stem cells from adult human fibroblasts by defined factors. *Cell*, 131(5):861–872, November 2007.

[27] Alfredo Tartarone, Giuseppina Gallucci, Chiara Lazzari, Rosa Lerose, Lucia Lombardi, and Michele Aieta. Crizotinib-induced cardiotoxicity: the importance of a proactive monitoring and management. *Future Oncology*, 11(14):2043–2048, July 2015.

[28] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *CoRR*, abs/1911.04252:10684–10695, 06 2019.

[29] Baichun Yang and Thomas Papoian. Tyrosine kinase inhibitor (TKI)-induced cardiotoxicity: approaches to narrow the gaps between preclinical safety evaluation and clinical outcome. *J Appl Toxicol*, 32(12):945–951, September 2012.

[30] Yi Yang and Peili Bu. Progress on the cardiotoxicity of sunitinib: Prognostic significance, mechanism and protective therapies. *Chemico-Biological Interactions*, 257:125–131, September 2016.

[31] Christian Zuppinger and Thomas M. Suter. Cancer therapy-associated cardiotoxicity and signaling in the myocardium. *Journal of Cardiovascular Pharmacology*, 56(2), 2010.

A Appendix A

Table 4: Hyperparameters used in experiments of iterative training and the most important hyperparameters for Mask R-CNN model. Some of them are tuned by simple grid search, but most of the values are as suggested in paper [16].

Parameter	**Value**	**Meaning**
score	[0.8, 0.9, 1.0]	value needed to filter mask for next training
backbone	ResNet50	backbone architecture of Mask R-CNN
img_per_gpu	6	number of images loaded per GPU
img_shape	[512,512,3]	shape of images to train on
l_r	0.001	learning rate
l_m	0.9	learning momentum
w_{decay}	0.0001	the decay of weights
L_{weight}	1	loss weights from multi-task loss are equal
num_classes	2	there are two classes - cell and background
epoch	50	number of epochs
steps_per_epoch	166	number of steps per epoch
val_steps	16	number of validation steps

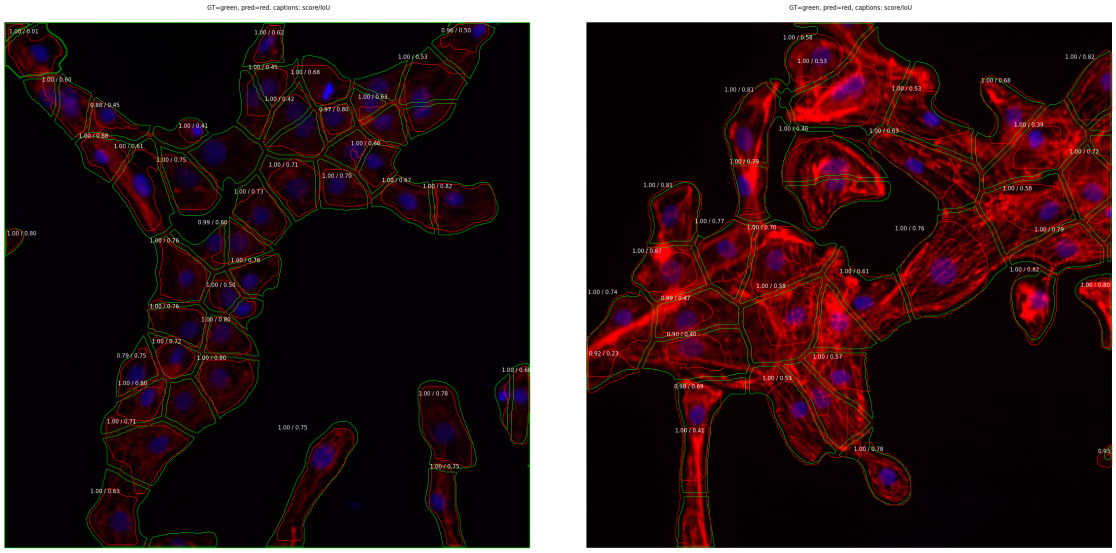
Most of the parameters in Table 4 are set to values recommended in [16]. Some of the parameters were tuned, such as learning rate l_r, number of epochs, steps per epoch or validation steps. We experimented with various values for the *score* parameter in the iterative learning model.

B Appendix B

In this section, images of predictions of the best models from each experiment are shown on chosen test data. The data were chosen according to the difficulty of detection for the model so that the differences between models are visible. We chose two samples that contain a high number of cells mainly clustered together.

B.1 Supervised Learning

Figure 7 shows the results from the best-trained model from the supervised learning experiments. The Mask R-CNN model was trained on 1000 samples of annotated data.

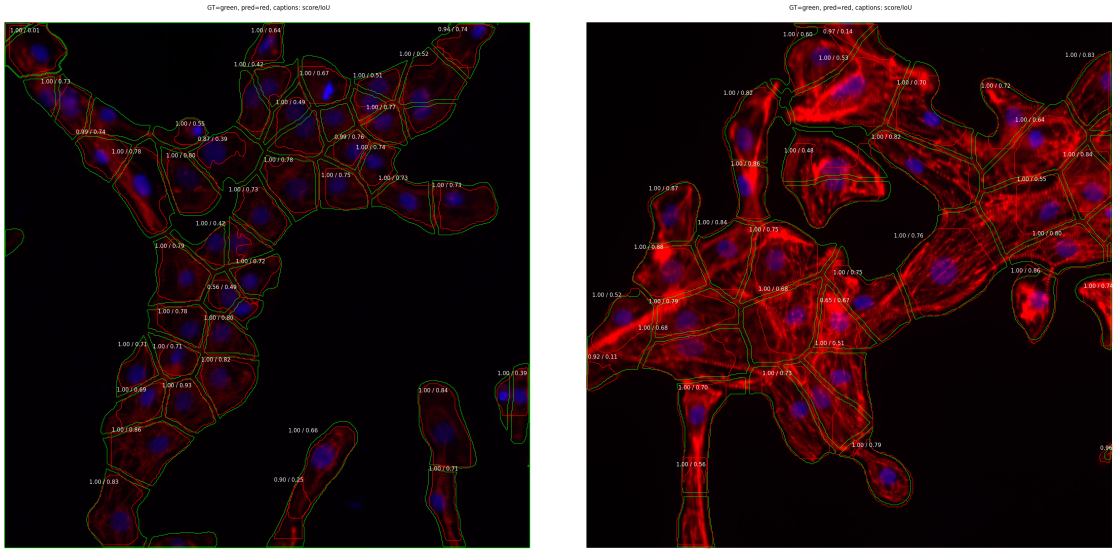


(a) (b)

Figure 7: Examples of predictions of Mask R-CNN trained on all annotated data. The green coloured lines note the ground truth masks, and the red coloured lines the predictions from the model. The score of the prediction and IoU value are also noted for each cell mask prediction [*score*/IoU].

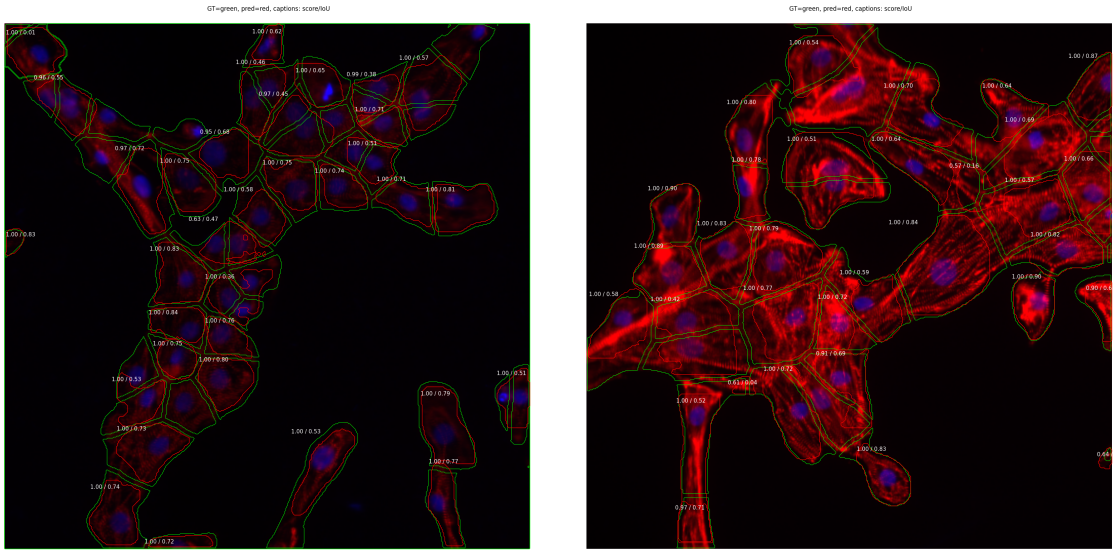
B.2 Semi-Supervised Learning

Figure 8 - Figure 11 show the results from the best-trained models from the semi-supervised learning experiments. Figure 8 shows examples of prediction from a model trained on 500 samples of annotated and another 500 samples of unannotated data, which is the first iteration of iterative learning. Figure 9 - Figure 11 show the examples of predictions from models trained in iteration IV of iterative learning. Each model used a different threshold value of the *score* to filter the masks - 0.8, 0.9 and 1.0.



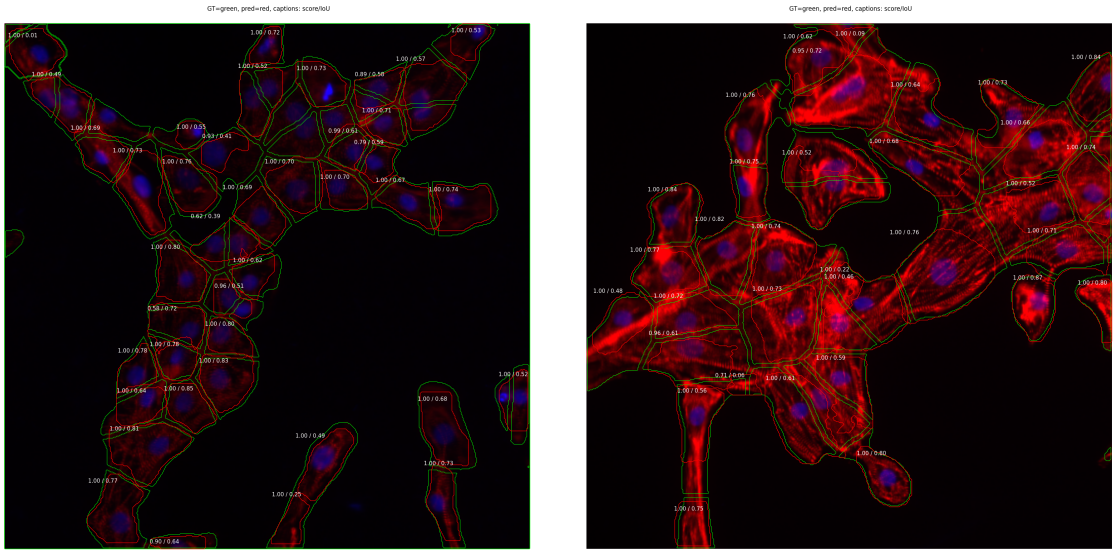
(a) (b)

Figure 8: Examples of predictions of Mask R-CNN trained on half of the annotated data and half unannotated. This is the first iteration of iterative learning. The green coloured lines note the ground truth masks, and the red coloured lines the predictions from the model. The score of the prediction and IoU value are also noted for each cell mask prediction [*score*/IoU].



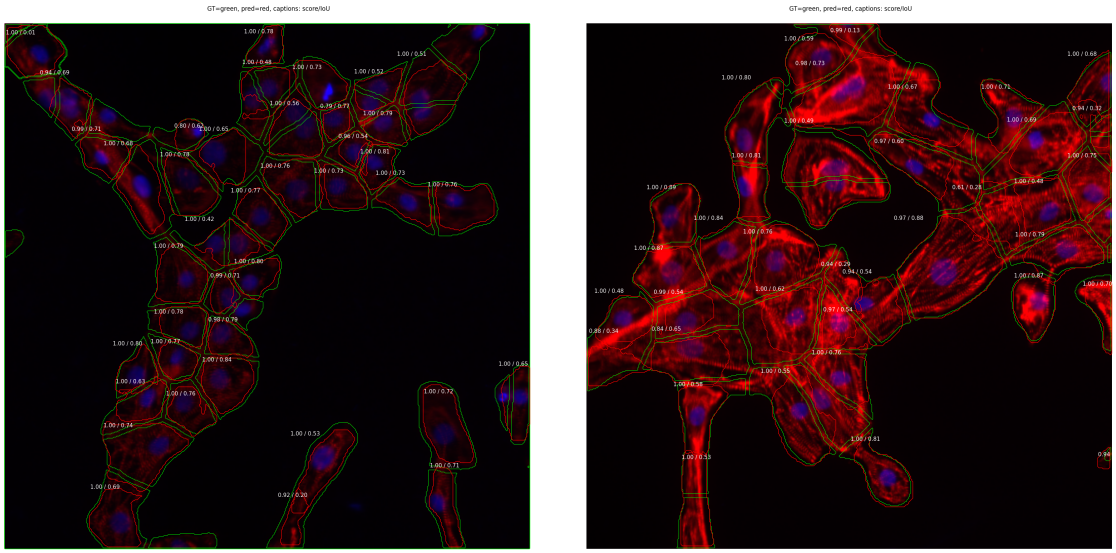
(a) (b)

Figure 9: Examples of predictions of iterative learning model with the *score* threshold for filtering the mask of 0.8. The predictions come from the model trained in iteration IV. The green coloured lines note the ground truth masks, and the red coloured lines the predictions from the model. The score of the prediction and IoU value are also noted for each cell mask prediction [*score*/IoU].



(a) (b)

Figure 10: Examples of predictions of iterative learning model with the *score* threshold for filtering the mask of 0.9. The predictions come from the model trained in iteration IV. The green coloured lines note the ground truth masks, and the red coloured lines the predictions from the model. The score of the prediction and IoU value are also noted for each cell mask prediction [*score*/IoU].



(a) (b)

Figure 11: Examples of predictions of iterative learning model with the *score* threshold for filtering the mask of 1.0. The predictions come from the model trained in iteration IV. The green coloured lines note the ground truth masks, and the red coloured lines the predictions from the model. The score of the prediction and IoU value are also noted for each cell mask prediction [*score*/IoU].