Machine learning Project 2

Wei JIANG, Xiaoyu LIN, Yao DI
EPFL, Switzerland

***Abstract*—The rapid development of deep learning technology triggers great improvements in traditional research area like image processing. This project is aiming at finding the best deep learning tools to segment the yeast cells from microscope image and support the researchers to accelerate their study about mathematical pattern behind cell division. Both the single object and multi-object segmentation algorithms will be applied for the research.**

I. Introduction

As one of the most addressed research area in image processing, image segmentation technologies have high potential to be applied in the various domains, including Content-based image retrieval(CBIR), medical imaging, biological forensics system and object detection.

Laboratory of the Physics of Biological Systems (LPBS) is focusing on discovering the mathematical principle behind the phenomena of interacting genes and cells. In order to quantitatively analysis the cell system, magnificent amount of data need to be collected. The resource of time required to process this amount of data with traditional methods is beyond the capability of the lab.

The project aims at applying the machine learning knowledge gained in the course to help the lab improve the research efficiency on yeast cell system. The report defines the research problem we are dealing with (Sec II), presents our method selecting processes and the principle of the selected algorithm (Sec III). Our efforts to adjust the standard model to the specific project are documented in Sec IV, including data prepossessing and algorithm tuning. The report also consists of a discussion (Sec VI) based on the results. The conclusion (Sec VII) are given at the end of the report.

II. Problem definition

The yeast cells are considered as one of the most suitable study materials for the cells interacting system, as they are unicellular, easy to inherit and do not generate harmful substance during their rapid reproduction process. In order to extract the information from the microscope images taken during the cell proliferation process, the first step is to extract each identical yeast cell.

Researchers from LPBS have already annotated several stack of images manually, which is a valuable data set for this project. We are going to find the suitable machine learning tools to learn from the annotated dataset and process the result from future experiments.

The objectives of the machine learning project include:

- Properly segment the yeast cell from the boundary.
- Properly segment the yeast cell from its neighbours.
- Identify the status of the yeast cells.

III. Methods Selection

Numerous image segmentation methods have been developed based on traditional image processing algorithms and widely applied, such as thresholding, edge detection, clustering or histogram-based methods. However, there is a gap between the performance of the traditional tools and the requirement from the latest computer vision applications.

In the past decade, several deep-learning-based algorithms have been created, in order to increase both accuracy and efficiency of image segmentation[1]. The major methods could be classified based on the applied deep learning technologies:

- Convolutional Neural Networks, e.g. U-net
- Convolutional Auto-encoders, e.g. Mask-RCNN
- Adversarial Models. e.g. Image to Image Translation
- Sequential Models. e.g. Conv. LSTM

We would like to take advantage of the following algorithms to accomplish our task and solve the yeast segmentation problem.

A. Single Object Supervised Learning: U-Net

The first method selected is the U-Net algorithm developed based on full conventional network (FCN) [2]. It is characterized by an auto-encoder with a mirrored series of convolution and max polling layers. The architecture of the U-net method could be found in Figure 1.

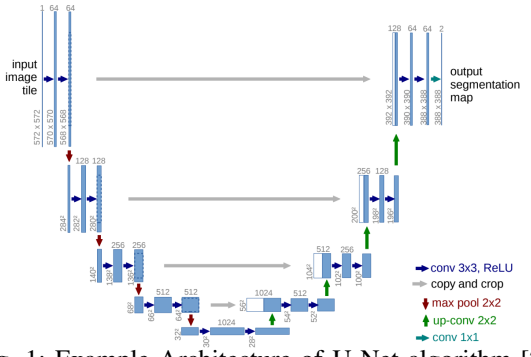


Fig. 1: Example Architecture of U-Net algorithm [2]

The method is chosen based on the following considerations:

- The U-Net method is a supervised-learning model for image segmentation. As we have the well-annotated dataset, it would be suitable to apply the supervised learning algorithm.
- The U-Net method was originally designed for biomedical image segmentation and it has been proven to have

good performance in cell tracking as the winner in ISBI cell tracking challenge. The method should be suitable for the current problem.

- The skip connector in U-net allows to copy uncompressed activation from the layer in encoder to the corresponding layer in decoder, which ensures the algorithm to generate crisp segmentation result.

B. Multi-object Supervised Learning: Deepmask & Sharpmask & Mask-RCNN

The obvious limitations of U-Net method is that it has only one objective, which means that the output of the algorithm will only be able to distinguish the cell pixels from the background. Post-processing of the binary segmentation map could not distinguish cells if they are too close to each other. Therefore, it is considered as a promising direction to apply multi-object supervised learning algorithms. With multi-object supervised learning algorithm, we could have the possibility to segment cells identically and even classify the cells with different statuses with single model.

The first two algorithms we tried are the Deepmask[3] and Sharpmask[4] algorithms developed by Facebook AI Research lab. The twin methods were both developed based on full convolutional neural networks with the additional procedure of multi-object mask determination and scoring. In terms of the differences, Deepmask considers segmentation as a classification process. It only forward extracts the features, classifies the pixels in high kernel and generates masks to low level. While Sharpmask changes the direction of the information flow in the deep net, using convolutional refinement to optimize the masks to have better boundary in the image level.

However, the first trials of these two methods could not provide us satisfactory results in terms of accuracy of cell recognizing.

The final model we selected as the representative of multi-object supervised learning algorithms is Mask-RCNN[5], which is developed according to Fast Region-based Convolutional Neural Network. The main architecture of the method is shown in Figure 2. It inherent from Fast RCNN method the ability of generate the classification score and region, as well as applying image segmentation skills from Deepmask to creating pixel-wise segmentation mask.

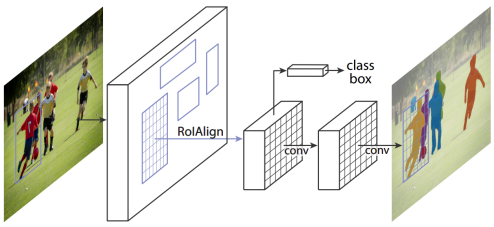


Fig. 2: Example architecture of Mask-RCNN [5]

IV. Data Prepossessing

A. General Processing for both methods

Fault annotation detection: Since the reference images are manually annotated by humans, there exits some mistake in those images. For example, no annotation in the last mask frame of `michael_1.2_im`, which will lead to some unexpected error and reduce accuracy. Before training our models, we should avoid loading those images. Also we observe some ground truths are not well-annotated which cause the oscillation of the loss. For example, in augoustina_first_im some parts of cells are not identified so we remove the incorrect region to get more accurate ones and then we find the loss decrease more smoothly.

Image flip: The manually annotated masks are expensive, so we only get numerous datasets for training. To make full use of the limited datasets and improve the performance of our models, we flip the images and corresponding masks in horizontally or vertically, or both horizontally and vertically to increase the size of datasets.

B. Special Processing for different methods

Image binarization: For U-net model, the ground truth of each cell is labeled with different pixel levels which is not reasonable since they belong to the same class. The idea is we transfer it into binary image with thresholding 0 to make training simple and then transfer back to multi-pixel image using watershed function in order to count the number of cells.

Image crop: When training with U-net model, we observe that most images are too large to train with our limited memory. So we crop them into smaller tiles which is also good for binary cross entropy loss since it performs bad with unbalanced proportion of background and objectives. We choose to use tile with overlapping to expand more images.

Instance extraction: For Mask-RCNN method, it is designed to generate segmentation masks for each instance of the objects in the image. For this specific project, there is only one class to detect, which is the cell, except backgrounds. We need to transform the given mask from one image to multi masks for each instance of yeast cell in the corresponding image. This process is easy because the different instance of yeast cell in the provided mask is given a different value.

Erosion & dilation : Another trick we use in the Mask-RCNN method is that we apply erosion augmentation before training. As the shape of yeast cell likes a circle, we use disk kernel for erosion, and set the erosion size to be the diameter of disk kernel. Then we apply dilation to the generated mask to improve the performance, with the same disk kernel.

V. Method Tuning and Results

A. U-net model

Our U-net model is built through Pytorch framework by which we can easily assemble a typical U-net structure. Limited by the computing power, we use four downsampling layers for encoder and four upsampling layers for decoder. In order to learn the edge information we use reflection as our

padding mode in convolution layers. We use BCEWithLogits as our loss function because its simple implementation for binary images. In terms of optimizer, we choose Adam because it is efficient and combines the advantages of AdaGrad and RMSProp. With several times tuning, a learning rate equal to 1e-5 is reasonable. We get 1206 image samples in total and randomly split them into training set and validation set(95% for training and 5% from validation).

During the training session, we find smaller tile size increase the accuracy since we get more tiles for training but it reduces accuracy. In order to solve the problem of limited memory and balance between efficiency and accuracy, we choose a tile_size equal to 512. So for every 2048*2044 image we get 64 tiles because overlapping with two times. Also considering the large area of background, the tile with cell area less than 36(based on our estimation of single cell area) will be excluded from training in order to make the process faster.

During validation session, we also tile the image to predict every part and put together back after prediction. We ignore the gradient to accelerate the validation process. By comparing the ground truth and prediction, we find the edges are not smooth. Besides, due to fact that the ground truth is usually larger than the cell itself, we dilate the threshold image with disk with diameter equal to 1. In order to count the number of predicted cells we use watershed function to label them with different colors. We used Gaussian blur topology to smooth color labeling in Watershed function. The following picture display the process of labeling.

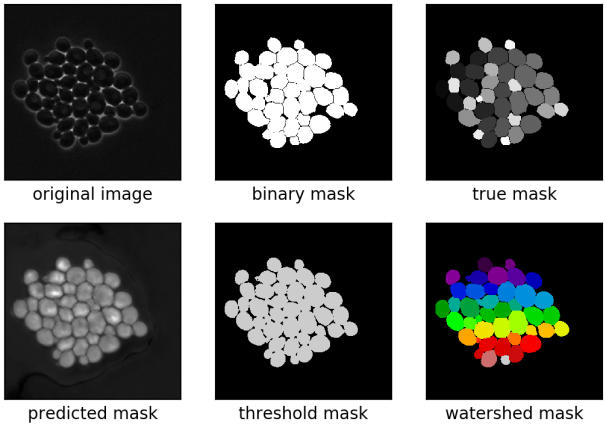


Fig. 3: Comparison between ground truth and prediction

The U-net model performs very well which we can see in the picture, some cells not annotated by human can be identify by U-net.

We observe that watershed sometimes separates a cell by mistake. In order to get better labeling, we have to pick good parameters for watershed: σ in and min_distance. The table below shows the parameter tuning by using the images in `michael_4.1_im.tif` as validation set and corresponding mask images as ground truth. As the table shows, we choose σ = 0.7 and min_distance = 6 as parameters.

TABLE I: Results watershed performance using different sigma and min_distance where the first column stands for ground truth.

σ	truth	0.5	0.5	0.7	0.7
min_distance	/	4	6	4	6
Number Fusions	0.00	0.38	0.67	0.38	0.72
Number Splits	0.00	1.33	0.71	1.38	0.67
Nb False Positives	0.00	0.95	0.71	0.90	0.71
Nb False Negatives	0.00	0.14	0.14	0.14	0.14
Average Overshoot	0.00	96.10	96.61	95.43	95.94
Average Undershoot	0.00	54.12	54.52	53.60	53.82
Av. True Area	1597	1602	1607	1603	1608
Av. Pred Area	1597	1645	1649	1644	1650
Nb Considered Cells	45.29	43.28	43.48	43.30	43.42

B. *Mask-RCNN*

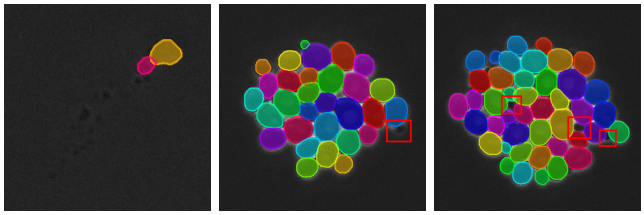
For Mask-RCNN method, we use TensorFlow framework for its `mrcnn` module. Also, because of the limitation of memory, we reduce the resolution of images from 2048*2044 to 1024*1024, and we also remove images of lower resolution. As mentioned in the previous sections, we use the image flip technique to extend the size of datasets. During each epoch of training, we randomly chose 100 images (iteration = 100, batchsize = 1) and randomly flip half of them horizontally, vertically or both horizontally and vertically. After training, we use 20 images for validation. Those validation datasets are not involved in train datasets to avoid overfitting problem.

We also apply the same technique in Mask-RCNN to improve performance. As the shape of the yeast cell is really like a circle, we generate a disk kernel for erosion and change its diameter (erosion size). We trained models using the different number of epoch and erosion size and evaluate their quality. For quality measurement, we also use the images in `michael_4.1_im.tif` as test dataset and corresponding mask images as ground truth. Because the test dataset is an image sequence, we average the results, which means some of the parameters are not so meaningful. The results of different situations are given in Table II. Because of the time-consuming training process, we set the maximum number of epochs to 10. From the evaluation results, we find erosion size of 3 and 10 epoch has better performance.

TABLE II: Results of Mask-RCNN using different erosion size and epochs for training: where ero means the diameter of erosion kernel, and ep means the number of epochs of training

Measurments	truth /	no ero ep=4	ero=5 ep= 4	ero=5 ep=10	ero=3 ep=10
Number Fusions	0.00	5.81	5.43	3.62	3.00
Number Splits	0.00	10.33	3.10	1.38	1.10
Nb False Positives	0.00	1.81	0.43	1.29	1.76
Nb False Negatives	0.00	5.57	5.43	3.57	2.90
Average Overshoot	0.00	106.12	54.15	41.30	46.89
Average Undershoot	0.00	189.65	291.86	286.68	232.29
Av. True Area	1597	1703	1763	1705	1686
Av. Pred Area	1597	1620	1525	1460	1501
Nb Considered Cells	45.29	30.76	37.10	40.29	41.05

By comparing the original test images and the generated mask images, we can easily observe the performance in more



Pollution problem Boundry problem Lower layer problem

Fig. 4: Some problem of generated masks using Mask-RCNN

detail. Figure 4 shows some problem of mask generated by Mask-RCNN method. The first image shows that the frame is polluted, and our model wrongly label that pollution as cells. The second figure shows that our Mask-RCNN model wrongly ignores some small inconspicuous cell on the boundary of the cell cluster. The third figure illustrates that the model wrongly decides the lower-layer cells as interval among cells.

VI. Discussion and Potential Improvement

A. Method Comparison

1) Accuracy: Based on the analysis documented in the previous section, we could easily observe that U-net method has better performance than Mask-RCNN in almost all dimensions except for Overshooting. Although Mask-RCNN has the potential to generate more interesting output, U-net is obvious the vanquisher in terms of segmentation quality.

2) Training Speed: In order to compare the training efficiency of the two different algorithms, we tried to train the two models with the same hardware set-up and the same image set. The average training time per image per epoch for the U-net method is 1.47s while the training time of Mask-RCNN reaches 43.25s. The high computation cost of Mask-RCNN is a main drawback for its future application.

B. Potential Improvement

Based on our experience, there are still several potential ways to further improve the performance of the selected method. These ideas have not been conducted due to the limitation of time and computation power. They could be utilized as reference in the future research.

From the aspect of the **dataset**: Prime training dataset is the prerequisite for generating a high-performance algorithm. In this project, it requires a great amount of resources to increase the quantity of the data while the annotation process is time-consuming. Therefore, it is not realistic to improve the dataset from the quantity side. The shortest board of the quality of the dataset is the annotation accuracy. We observed several annotated mask with missing or wrong annotated cells. It could be improved by 4-eye checking between the raw image and the annotations. Secondly, multi-object labelling for cells in different phases could make the dataset more valuable for future research.

From the aspect of **tuning current model**: in U-net model, it is possible to implement dilated convolution layer rather than standard convolution to get more detailed information and avoid information loss. Also, we can build a more suitable loss function to learn the edges well by considering the imbalance between background and objectives, such as Dice loss or the weight map function to give the position of cells more weight loss.

From the aspect of the **merging both models**: There is a promising direct to merge U-Net and Mask RCNN method, i.e. apply the skip connectors to transfer the info from convolution layer to max-polling layer during the mask generation process of Mask RCNN model. So that the final model could have the advantages of both method: identify different objects as well as maintain segmentation quality.

VII. Conclusion

During this project, we managed to use the machine learning related knowledge learnt in the course to segment yeast cells in microscope images. Two different approaches have been applied, including single object approach with U-net algorithm and multiple object approach with Mask RCNN model. Future improvement could be done in by improve the quality of dataset, tuning current model and creating new models.

Acknowledgements

We thank Prof. Sahand Jamal Rahi for providing us the opportunity to work on this project. Many thanks to Prof. Jaggi Martin, Prof. Urbanke Rüdiger and all the teaching assistants in the CS-433 course for bring us the knowledge in machine learning area. Grateful acknowledge to the help from Matthias Minder and the following git contributors. The codes generated by them provide a good starting point for this project.

- Unet [6]
- Deepmask/Sharpmask [7]
- Mask-RCNN [8] [9]

References

[1] S. Ghosh, N. Das, I. Das, and U. Maulik, "Understanding deep learning techniques for image segmentation," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, p. 73, 2019.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[3] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[4] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*. Springer, 2016, pp. 75–91.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[6] milesial, "Pytorch implementation of the u-net for image semantic segmentation with high quality images," https://github.com/milesial/Pytorch-UNet, 2019.

[7] N. Zenovkin, "Tensorflow implementation of deepmask and sharpmask," https://github.com/aby2s/sharpmask, 2017.

[8] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[9] E. Arnaudo, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/edo-arn/sharpmask-mrcnn, 2018.