

# Semantic segmentation of L-forms

Ismoil Odinaev

The Chinese University of Hong Kong,  
Faculty of Science, Department of Physics

Final Year Project (Phys4620)

Spring 2021

## Abstract

This project deals with semantic segmentation of biological organisms called L-forms. Segmentation is done here in two ways: model-based in MATLAB and Deep Learning model in Python. The Convolutional Neural Network model used is called U-Net. A comparison between the two approaches is given.

## **1. Introduction**

Enhancement of Artificial Intelligence and Machine Learning algorithms provided scientists with powerful tools. It enabled researchers to improve efficiency and diminish research time, opening and discovering the road to other discoveries. This paper will present semantic segmentation of biological organisms called L-forms by using Convolutional Neural Networks (CNN) and MATLAB modelling. First theoretical background and algorithms are explained, then results are presented.

## **2. Theory**

### **a) What is L-forms?**

L-form is bacteria, also known as cell wall deficient bacteria, a phase of microscopic bacteria and lacks cell walls. Last 100 years, lots of research was done on L-forms; however, they remain primarily misunderstood - or at the very least, underappreciated by the medical research community. L-forms might be the causes of some chronic diseases. L-form bacteria are pleomorphic, i.e., they change size and shape. During much of their lifetimes, they are tiny, about 0.01 microns in diameter. L-forms cannot be seen with a regular optical microscope. The small, individual forms of L-form bacteria are often referred to as coccoid bodies. Coccoid bodies sometimes group together, assuming the appearance of a string of pearls. [3]

### **b) Different methods of detection of biological organism**

Image analysis is heavily used to quantify phenotypes of interested objects. Images are also being used as unbiased sources of quantitative information about cell state in an approach known as image-based profiling. In image-based cell profiling, hundreds of morphological features such as shape, size, intensity, and contrast are measured from a population of cells. The workflow of image-based profiling can be divided into four steps, which can be divided into sub-steps. These four steps are Feature Extraction, Data Preparation, Profiling and Analysis. [1]

Each of these four steps can be done in numerous ways. For example, Feature Extraction firstly can be done by the process called segmentation. Segmentation is a process where each cell in the image is identified and measured individually; that is, its constituent pixels are grouped to distinguish the cell from other cells and from the background. There is model based approach and machine learning approach. Model-based approaches typically involve histogram-based methods, such as thresholding, edge detection, and watershed transformation. On the other hand, machine learning approach trains a classifier to find the optimal segmentation solution by providing it with ground-truth data and manually indicating which pixels of an image belong to different classes of objects. [1]

Similarly, other steps can be done using various statistical tools and models. However, here we deal mostly with segmentation: model-based in MATLAB and Deep Learning (CNN) in python.

### **3. Computational part**

#### **a) MATLAB**

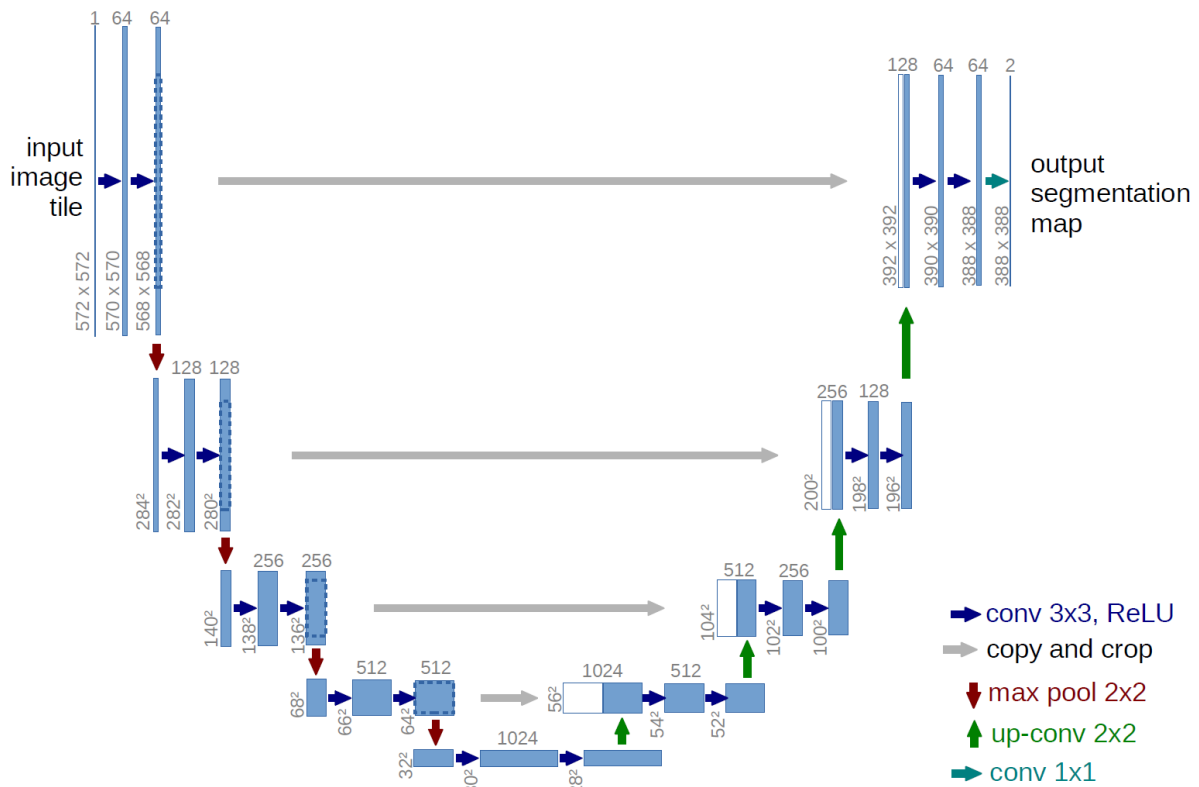
MATLAB modeling is done via Image Processing Toolbox and Computer Vision Toolbox. First process is edge detection. Then morphological operations follow (a technique to analyze structure, shape, and sizes): structuring element, dilation, and finally filling. Structuring element is a shape, used to probe or interact with a given image, with the purpose of drawing conclusions on how this shape fits or misses the shapes in the image [4]. Then dilation operation uses a structuring element for probing and expanding the shapes contained in the input image.[5] The final step is filling morphological structures that have been created in previous steps.

#### **b) Convolutional Neural Networks**

Convolutional Neural Networks (CNN) is a class of deep neural network, most commonly applied to analyze images and videos. CNN accept tensors as inputs and for images it is 4-dimensional tensor which is (number of inputs) x (input height) x (input width) x (input channels). Number of channels is mostly 3 due to RGB and 1 when image is grey scaled. CNN mainly consist of convolutional, activation and pooling layers. Convolutional layers can be assumed as filters which detect various features of image and together with activations (it should be noted that activations are needed to detect non-linearity in the features), they result in feature maps which contains detected features. Initial layers detect features like horizontal, vertical lines or edges. Subsequent layers detect more complex features like shapes such circles or squares and for deeper model they can detect even more complex features. Then, pooling is needed to highlight most present feature. Here U-Net model is used which was developed by graduate students from University of Freiburg in 2015. [2]

#### **U-Net**

U-net was designed for biomedical image classifications and it works particularly well for this purpose, because this model can perform outstandingly even with small amount of data which is the case for biomedical tasks. U-Net is written on TensorFlow in Python in this project.



The architecture contains two paths. First path is the contraction path which is used to capture the context in the image. The contraction path is a stack of convolutional and max pooling layers. In the first part, the size of the image gradually reduces while the depth gradually increases. What happens here is network learned features (or information) in the image, however it has lost the “position” of the information. The second path is the symmetric expanding path which is used to enable precise localization using transposed convolutions. In this part the size of the image gradually increases as the depth gradually decreases. So, expanding path recovers the “position” of the information. To get precise locations, at every step of the expanding path, the output of the transposed convolution layers is concatenated with the feature maps from the contraction path at the same level. Furthermore, dropout layers are added in each step. These layers randomly drop some parts of layer in each iteration and hence, prevent overfitting and learning of unnecessary noise.

Finally, it should be mentioned that U-net is an end-to-end fully convolutional network i.e., it only contains Convolutional layers because of which it can accept image of any size.

## Data Augmentation

This is one of the first steps before training model if one does not have enough images. For this project initially 113 images were provided which is not sufficient for training. Albumentations library in Python was used for that purpose. In this step images are copied with different changes such as adjusting contrast, rotating the image by  $90^\circ$ , cropping images, horizontal/vertical flips, etc. The most important for biomedical images is optical distortion and grid distortion because biomedical images deal with non-rigid structure with various shapes [6].

## Compiling the model

The model compiled with loss function is binary cross entropy. Binary entropy is given by:

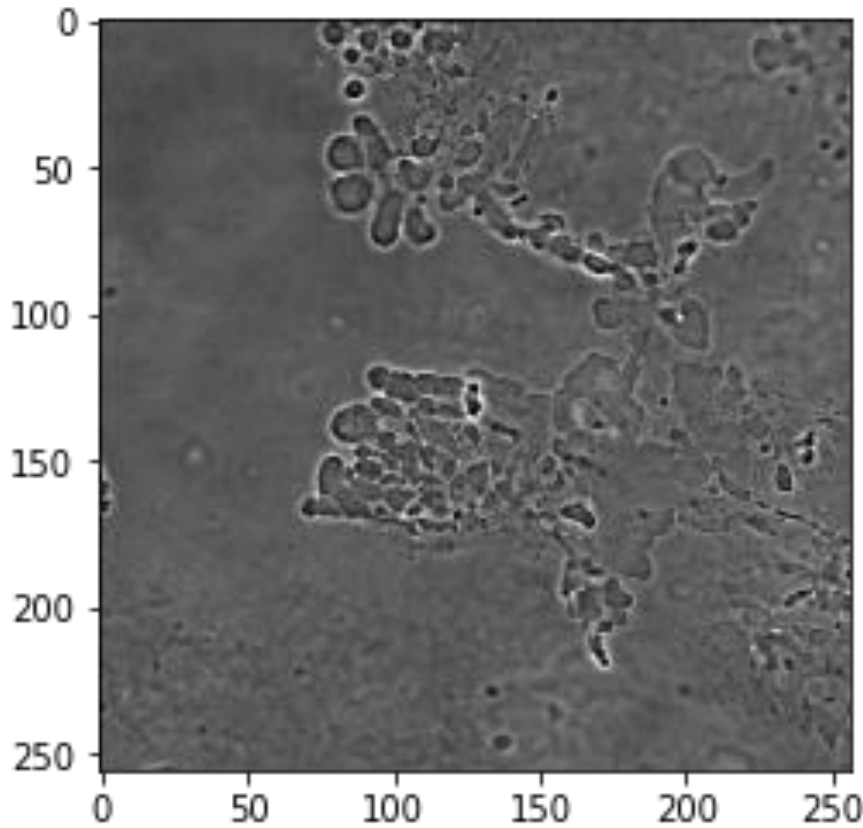
$$H_{PQ} = -\frac{1}{N} \sum_{i=1}^N y_i \log p(y_i) + (1 - y_i) \log (1 - p(y_i))$$

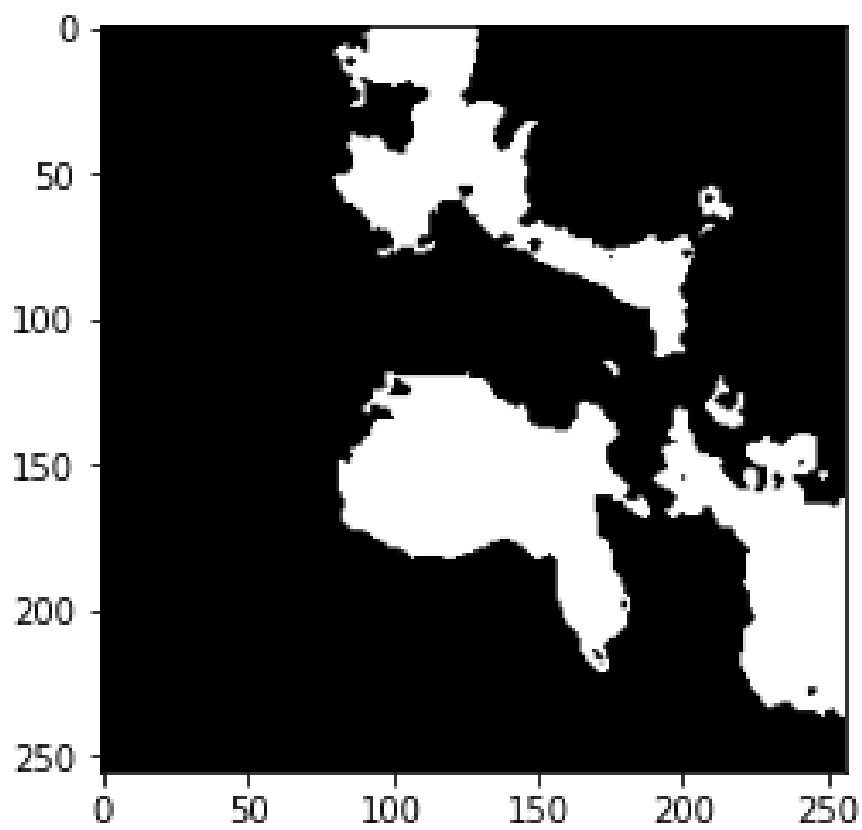
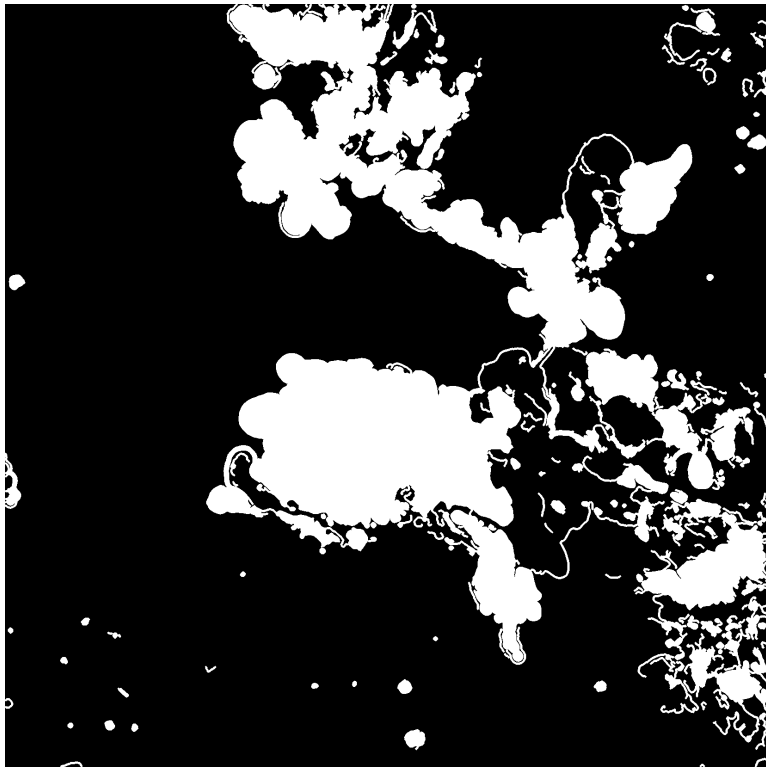
where  $y_i = 1$  to one class and  $y_i = 0$  refers to another. The lower entropy the better results. Then, optimizer ADAM is used which is a stochastic gradient descent algorithm with adaptive learning rate for each parameter. Finally, data is split into two parts: 85% is for training the model also called training data, 15% is for validation, which is used to see the performance of the model.

#### 4. Results

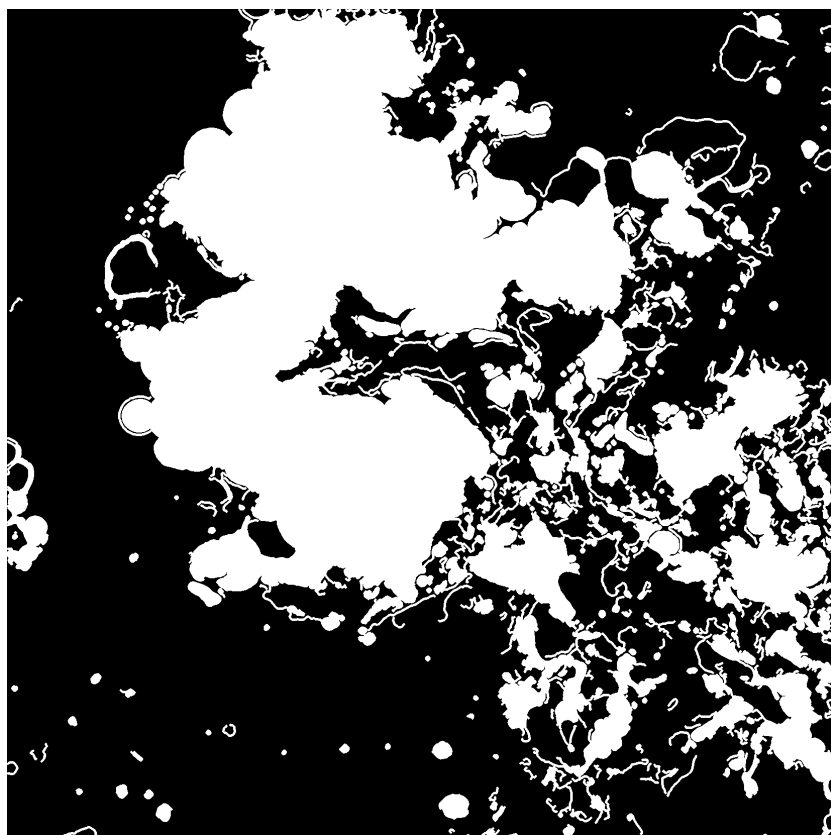
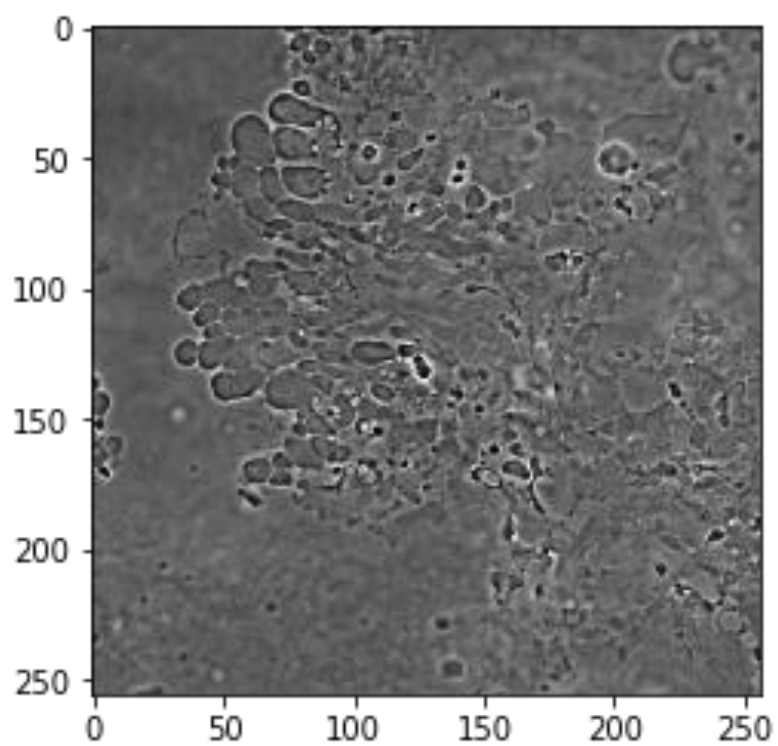
Validation accuracy is 89.75% which is very good, considering that there were small amount of data. Here we present some results from both MATLAB and U-Net. (First is testing data,

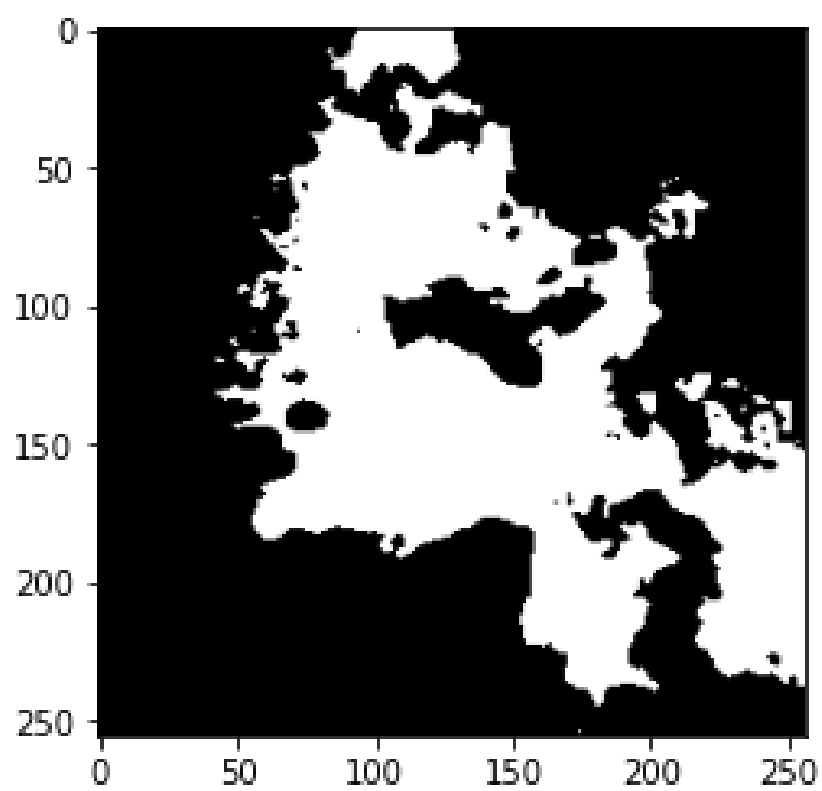
second is MATLAB,  
third is U-Net)



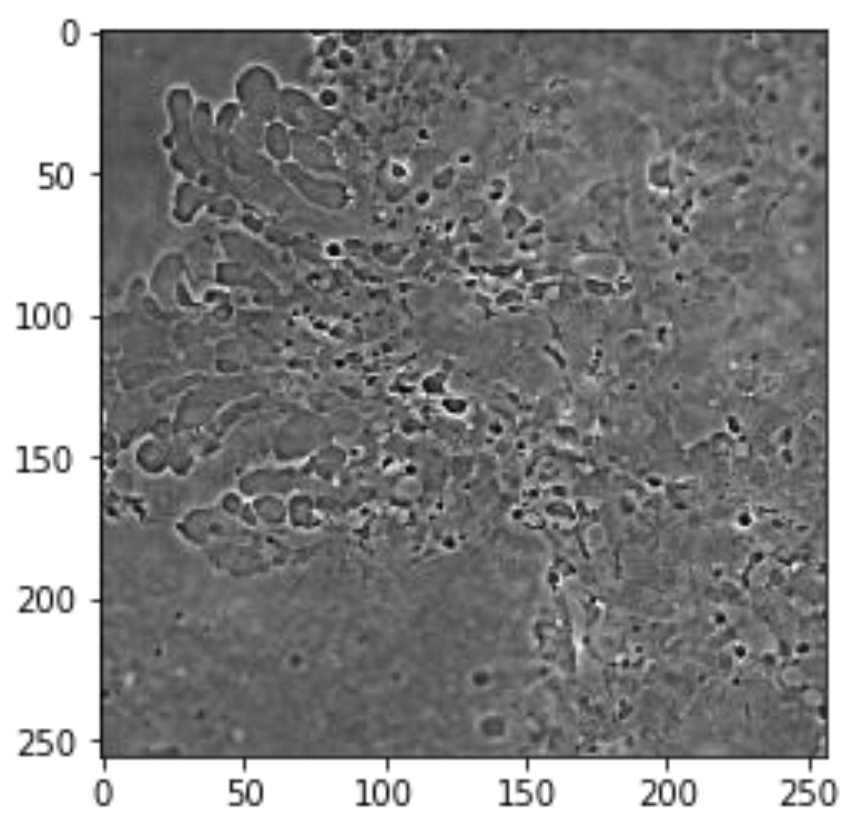


And some other images:

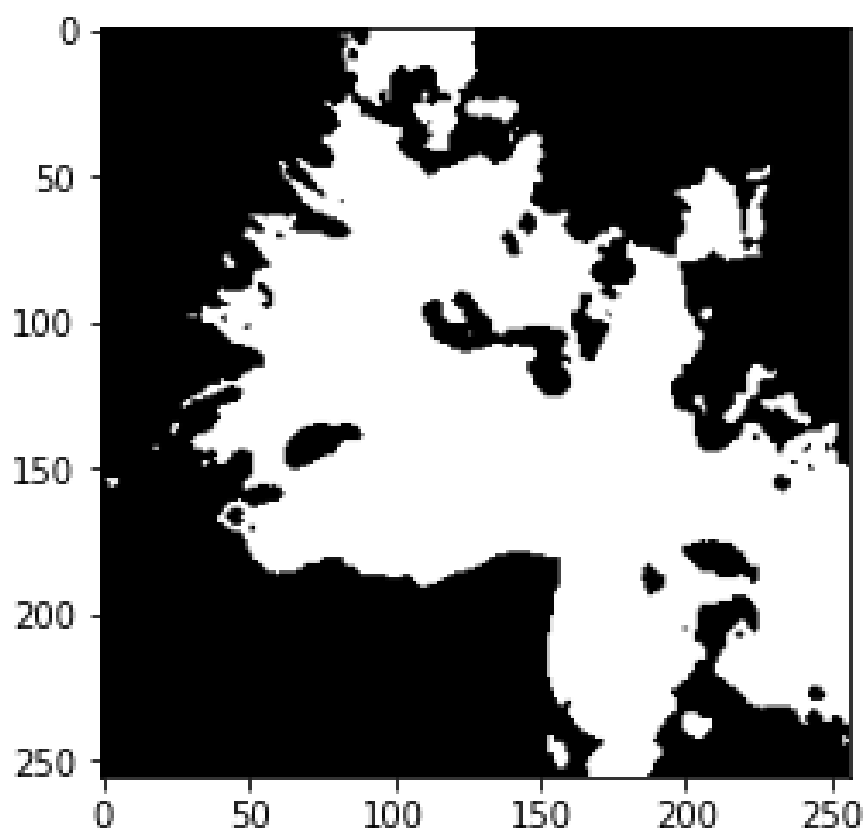
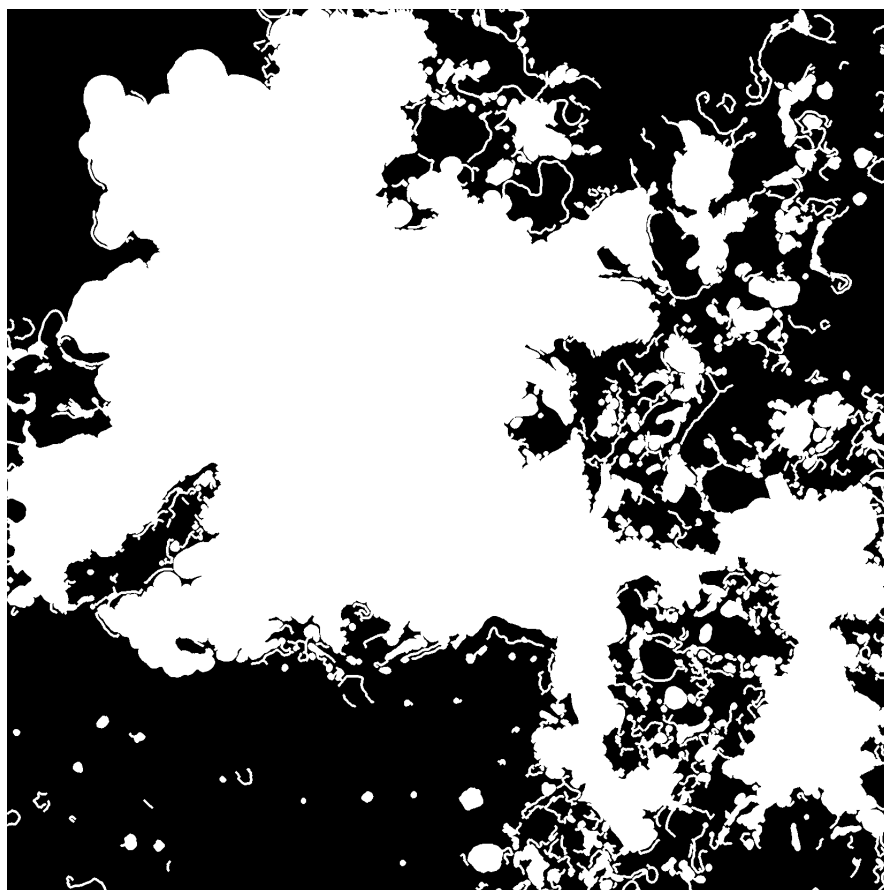




And last images:







## 5. Discussions:

While both approaches are used in profiling experiments the model-based approach can perform well only on simple segmentation tasks. Model-based approach requires manual parameter adjustment for each new experimental setup and one setup does not work for all shapes and sizes. (see the images: for complicated shapes, model-based perform poorly because one code was used for all three and no adjustments were done for different images) However, Machine-learning-based segmentation can perform far better on difficult segmentation tasks, (~90% accuracy) such as highly variable cell types or tissues. It does not require as much computational expertise, but it require labeling of training pixels for each experimental setup.

Furthermore, instance segmentation can be done to detect every instance of the desired object. However, to do that one need manually prepare training dataset by spotting every instance separately which consumes lots of time.

## 6. Appendix

All source codes are provided.

### *References*

1. Data-analysis strategies for image-based cell profiling, Nature Methods, Vol14, No:9 (2017)
2. U-Net: Convolutional Networks for Biomedical Image Segmentation; Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015)
3. [L-form bacteria \(MPKB\)](#)
4. [Structuring element - Wikipedia](#)
5. [Dilation \(morphology\) - Wikipedia](#)
6. [\(PDF\) Albumentations: fast and flexible image augmentations \(researchgate.net\)](#)