Universiteit Leiden

Master Computer Science

Image segmentation using regression models, fuzzy c-means clustering, and deep learning methods.

Name: Yijia Li
Student ID: s2528622
Date: 17/08/2021

Specialisation: Bioinformatics

1st supervisor: Dr. L. Cao
2nd supervisor: Dr. K.J. Wolstencroft

Master's Thesis in Computer Science

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

Abstract

Anthracycline chemotherapy is an important treatment to tumors, but this drug has a serious side effect of leading to heart failure. The research on cardiotoxicity requires the culturing of cardiomyocytes, and GFP is used to indicated the cardiomyocytes out of the stem cell line. Efficient ways of analyzing fluorescent images becomes essential in this experiment. However, GFP signals vary between different batches of cells. Well-established automatic thresholding tools in ImageJ cannot satisfy the need of finding fluorescent signals from images.

In this thesis, we first try to use the gray value features of images to predict the appropriate binary thresholds. The experiment shows obvious underfitting and no bias on gray value features. Then, we try unsupervised learning method, the fuzzy c-means clustering, to see if clustering methods can have a good performance. This method shows good results on some of the images, but cannot handle the images with too many or too few cells in vision. At last, we use U-Net as an example of deep learning segmentation method to segment the image into foreground and background. We try different sizes of input images, and experiment with the biggest size of input images (1024x1024) shows the best performance, with a test set accuracy of 0.7556 and IoU of 0.8468.

Keywords: cardiotoxicity, thresholding, machine learning, fuzzy c-means, U-Net

Acknowledgements

Firstly, I would appreciate to both of my supervisors, Dr. L. Cao and Dr. K.J. Wolstencroft, for their help, advice, and comments on the thesis. They not only help me in thesis field, but also bring me happiness in this special period of time.

Secondly, I would appreciate to all creators and developers that I mentioned in thesis.

Thirdly, I would thank my friends that I met in Leiden. It is them who help me through bad emotions and problems.

At last, I would thank my families who keep supporting and pushing me.

Content

Chapter 1 Introduction......6
1.1 Research on cardiotoxicity......6
1.2 Related works in image segmentation......6
1.2.1 Classic segmentation method......6
1.2.2 Machine learning method......8
1.2.3 Deep learning based method......8
Chapter 2 Methodology and experiments......10
2.1 Dataset and problem......10
2.2 Threshold prediction using machine learning methods......11
2.2.1 Features chosen......11
2.2.2 Problem formulation......12
2.2.3 Machine learning methods......12
2.2.4 Data pre-processing......13
2.2.5 Experiment......13
2.3 Fuzzy C-means......14
2.3.1 Algorithm......14
2.3.2 The pseudo code of Fuzzy C-means algorithm......15
2.3.3 Experiments......15
2.4 U-Net......15
2.4.1 The principle of U-Net neural network......15
2.4.2 Data pre-processing and train-test set chosen......16
2.4.3 Network initialization......17
2.4.4 Training and evaluation......17
Chapter 3 Results......18
3.1 Threshold prediction using machine learning methods......18
3.2 Fuzzy C-means algorithm......19
3.3 Segmentation using U-net......22
Chapter 4 Discussion......26
4.1 Threshold predicting using machine learning method......26

4.2 Fuzzy C-means algorithm 26
4.3 Segmentation using U-Net 27
4.4 Other...27
Chapter 5 Supplemental Materials..28
Reference 32

Chapter 1 Introduction

1.1 Research on cardiotoxicity

Tumors come to be one of the biggest risks to human health. Among many tumor treatment methods, anthracycline chemotherapy is an important one, with an effective performance on many kinds of cancers. However, one side effect it brings is the cardiotoxicity. The most serious outcome could be heart failure[1].

Heart functions as the pump which pushes the blood to every corner of the human body in order to transport oxygen, nutrients, and metabolic waste. Human heart has four chambers separated by four valves. The one-way valves limit the direction the blood flows. Human heart wall is a three-layer structure: endocardium, myocardium, and epicardium, from inner to outer. The inner endocardium is connected with the chambers and valves. The outer myocardium functions as the protection and lubrication. When these muscles contract in an appropriate and coordinated way, the ventricle is twisted and squeezed from several directions, and blood flows out of the heart. Sarcomere is the fundamental unit of heart muscle cells, or cardiomyocyte, and it contributes a lot in the contraction of the heart muscles. The change in sarcomere leads to the change in contraction capability of the heart.

Cardiac remodeling is considered as a cause of heart failure as well as many other kinds of heart diseases. In cardiac remodeling, the size of cardiomyocyte increases, and its shape changes. Cardiac wall stiffness increases in this process[2] by changing the properties of sarcomeres in the heart. However, the detailed mechanisms are not very clear so far. Research on heart muscle cells and efficient observation methods are very essential.

Human stem cells and derived cardiomocytes are popular material in heart study. To better analyze the performances of certain proteins, green fluorescent protein (GFP) is well used as a fusion protein to indicate the location of target proteins. GFP shows green fluorescence when it is exposed to light from blue to ultraviolet, which is useful in labelling target proteins. Since the targets of interest have been labelled with green fluorescent, the following step is to find an efficient way to segment these pixels from the background of a fluorescent image.

1.2 Related works in image segmentation

1.2.1 Classic segmentation method

In the field of computer vision, image segmentation refers to the process that separates the whole image into multiple sub-areas. All of the pixels inside one sub-area are expected to indicate some characteristics of the original image, such as color, texture, or distinguish the foreground out of the background. There are various methods for image segmentation.

Thresholding is the simplest method for grayscale image segmentation. The main purpose of thresholding is to choose a fixed value as the threshold, and separate the foreground and the background by the gray values of each pixels.

For digital biological images, ImageJ is a very popular software used for image processing and analysis. In addition, it encompasses many classic auto thresholding algorithms[3]. We firstly give a brief introduction of those widely used thresholding algorithms.

IsoData

At the very beginning, an initial threshold is set randomly. Then, the image is divided into the foreground and the background. Subsequently, the mean gray values of foreground and background are computed. The new threshold is replaced by the average of the two mean gray values. The process is repeated several times until convergence[4].

Intermodes

This algorithm is based on the histogram of gray values. The histogram is firstly assumed to be bimodal, and then smoothed several times until two wave peaks are achieved. The threshold is set to the mean value of two peaks. However, this method does not work well while the image histogram is not bimodal, or the wave peaks are not distinct[5].

Otsu

In Otsu[6] algorithm, the image is divided into two classes, the background and the foreground. All thresholds available are traversed and are used to calculate the within-group variances. The threshold with the least within-group variance is decided as the threshold for binary segmentation.

Seeded region growing

In seeded region growing (SRG) algorithm[7], the seed points are firstly selected according to some features, such as gray values. Each seed point refers to certain area of the image. Then, the area grows with the seed point as origin, and the nearby pixel points are be merged based on certain criteria, such as texture similarity. However, SRG could be too sensitive to noise of certain pixel, which brings a major disadvantage of this algorithm.

Watershed Separation

In watershed algorithm[8], the whole image is treated as a map, in which the gray value is considered as the height of each pixel, and the threshold can be seen as the water level. The water level increases from the minimum of the gray value until it overflows, and dams are set in case the water merges. The segmentation is accomplished according to the lines of pixels where dams lie.

Edge detection

The aim of edge detection[9] is to find pixels where the grayscale value changes intensely. The detection could be achieved by search-based method and zero-crossing based method. In search-based method, different gradient is used to estimate the direction of edge and find the local maximum. In zero-crossing based method, the edge is positioned by zero-crossings of non-linear differential expression such as the Laplacian.

1.2.2 Machine learning method

One attempt on supervised learning is using K-means clustering. In K-means algorithm, the number of clusters k is determined, and centers are initialized. The Euclidean distances between every cluster centers and every pixel are calculated, and each pixel is assigned to the near cluster according to the distance. Since all pixels are assigned, the new centers of all clusters can be calculated from the pixels, and then the new distances between centers and pixels are recalculated. This process is repeated several times until convergence, and the output segmented image is achieved[10]. A lot of works are used to improve the performance, such as using partial contrast stretching to enhance the contrast[11], using subtractive clustering to find optimal initialization centers[12].

Support vector machine (SVM) can be also used in the image segmentation. In SVM, a hyperplane $w \cdot x + b = 0$, separate the points in the dataset into two parts. For $y_i = +1$, $w \cdot x_i + b \geq 1$. And for $y_i = -1$, $w \cdot x_i + b \leq -1$. The purpose of SVM is to find a hyperplane with the minimal distances. SVM shows good performance in image segmentation field, such as road extraction[13] and medical analysis[14] .

1.2.3 Deep learning based method

There are a lot of successful attempts of deep learning methods in the field of image analysis. In computer vision field, convolutional neural network (CNN) is one of the most popular deep learning methods[15]. CNN is usually made up of three parts[16]. The first one is the convolutional layer, in which the convolutional kernel slides over the image and extracts the image features. A feature map appears after this step. In the second part, down-sampling layer is used to reduce the dimension. In this step, the number of parameters to be trained is decreased so as to enhance the running speed and reduce the overfitting. The third part is the fully-connected layer. Here, all feature maps are linked together and final classification function is used. AlexNet[17] is a very early and successful application that using CNN in computer vision field. It consists five convolutional layers followed by three fully connected layers. Instead of sigmoid or Tanh that is used in CNN before, in AlexNet ReLU is used to solve the gradient disappearance problem. Dropout is also used in AlexNet to reduce the overfitting.

VGG[18] is also successful in image processing filed. Compared to AlexNet, it uses several small convolutional kernels instead of the bigger ones in AlexNet. In short, VGG enhances the image processing capability by consuming more computing resource in trade-off. Among all networks sharing similar structures, VGG-16 and VGG-19 are both the popular ones. You can easily find pre-trained VGG-16 and VGG-19 networks for convenience.

Although CNN is very successful in image classification tasks, one challenge for recent well-used CNN is the information loss, due to the multiple convolutional processes and down samplings. In this case, the feature map with small size can be very efficient in classification tasks, but could also be too small to store enough image information[19]. Fully convolutional network (FCN)[20] is a good solution to this problem. Instead of the final full connected layer of most CNN, FCN uses one up-sampling layer after the final feature map. The size of the feature map after this step is restored the same as the original image, making it possible to analyze pixel by pixel and solve more tasks apart from classification. Another solution is the encoder-decoder architecture network, such as U-Net[21]. Its nearly symmetry U-shape comes from an encoder part and a decoder part. The encoder part is also called the contracting path, which consists multiple convolutional and maxpooling layers. The decoder part, or the expanding path, using deconvolution to restore the feature to the original sized segmentation map.

Chapter 2 Methodology and experiments

2.1 Dataset and problem

Fluorescent photographs are very popular in biomedical research, because fluorescence probe is well-used as indicators. However, several properties of fluorescent images suggest that it is not easy to find the best way to analyze them. First, it requires a dark field for taking fluorescent images. Blurring and fuzzy are both common disruptions, because it is not very easy to take a good fluorescent photo. Focus is a common reason. Another important thing is to choose the appropriate exposure time. Second, it is hard to always remain a consistent environment while making slices or taking images for monolayer beating cells. The fluorescent signal strength of biological samples varies a lot between different batches. It is certain that the images of different batches are have very diverse properties, which makes it even harder to analyze. The unevenly thickness of samples and coverslips is another cause.

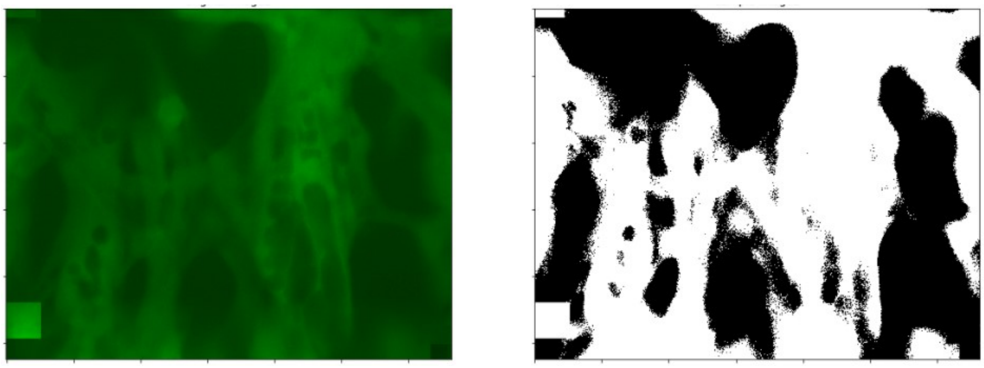
The dataset of this project is 91 fluorescent images of green fluorescent protein tagged cardiomyocytes. This image dataset incorporates different treatment conditions such as varied drug concentrations and treatement days. The images from red fluorescent channel and brightfield are not be used for this project. GFP images are formatted as three channels (RGB) 8-bit color graphics. Only the values in green channels are kept. The red and blue channels are empty. The size of the image is 1328×1048. For each fluorescent image, it comes with a binary mask image, which are binarized according to manually set thresholds by biologists. These binary masks are treated as ground truth data set for performance evaluation and deep learning model training. The purpose of this project is to better separate the foreground (the GFP positive cells) from the background of each image, since traditional thresholding tools for biologists, such as ImageJ, do not always bring satisfying results. In these images, fluorescent signals are not evenly distributed, and cells lie in multi layers and do not scatter uniformly.

In this project, we first use machine learning methods to explore a potential way of predicting a fixed threshold based on the properties of image histograms. Also, we use Fuzzy C-means as an example of clustering algorithms. Then we use U-Net neutral network, to see if artificial neutral network could better solve this segmentation problem.

Some examples of the dataset can be seen in Figure 1, including the gray value features as well as the original image and binary mask.

		Area	Mean	StdDev	Mode	Min	Max	IntDen	Median	Skew	Kurt	RawIntDen	thre
0	1	1391744	57.662	30.694	25	0	251	80250320.0	55	1.004	1.304	80250320.0	45
1	2	1391744	51.750	30.758	32	0	248	72022392.0	40	1.980	4.607	72022392.0	53
2	3	1391744	54.572	24.479	39	0	248	75950564.0	47	1.069	1.875	75950564.0	54
3	4	1391744	56.898	26.520	41	0	246	79187225.0	51	1.429	2.434	79187225.0	61
4	5	1391744	81.965	52.599	24	0	251	114074656.0	71	0.647	-0.303	114074656.0	38

Gray value features



Original image　　Binary mask

Figure 1. Some examples of the dataset

2.2 Threshold prediction using machine learning methods

2.2.1 Features chosen

We firstly try to predict the threshold of an image based on features from gray value. ImageJ provides a convenient way of extracting the image features for this experiment. The features we choose are shown as in Table 1.

Feature	Explanation
Mean	The average gray value of every pixel
StdDev	Standard Deviation The standard deviation of gray values
Mode	Modal Gray Value The gray value of the biggest frequency
IntDen	Integrated Density The sum of gray values of all pixels
Median	The median gray value of every pixel
Skew	Skewness The third order moment of the average gray value
Kurt	Kurtosis The fourth order moment of the average gray value

Table 1. Features of images

2.2.2 Problem formulation

With 91 images, each having 7 feature values, we are trying to use data

$$(x_1, x_2, .. x_7)^{91}$$

to predict the thresholds

$$(y_1, y_2, .. y_{91})^T$$

using different machine learning algorithms.

2.2.3 Machine learning methods

The images are all 8-bit, with a continuous value from 0 to 255. We firstly want to make use of different regression models instead of classification models. We try different machine learning algorithms in this prediction task. All the regression tasks are based on *Scikit-learn* package[22] in *Python*.

Linear

Linear regression uses linear regression equation to accomplish the prediction of variables. In this task, we use multiple linear regression to predict the threshold using values of different image features.

Logisitic

In logistic algorithm, a sigmoid function is added in the linear regression model. Furthermore, it uses maximum likelihood method, instead of least squares method in linear regression.

Polynomial regression

In polynomial regression, n^{th} degree polynomial model is used to predict the dependent variable based on the independent variable. Thus it could be regarded as a special linear regression model.

Ridge and Lasso regression

Ridge and Lasso regression are both special cases of multiple linear regression models. They use different regularization methods to solve the overfitting problem. In ridge regression L2 regularization is used, and L1 regularization is used in lasso regression.

Bayesian linear regression

Bayesian inference is used in this model. The parameters of the linear regression are regarded as random variable, and posterior is calculated from the prior.

2.2.4 Data pre-processing

Outlier detection is a very important step in machine learning, because outliers make the model less fitting and accrate. Z-score test is used in the pre-processing step.

$$Z_i = \frac{x_i - \mu}{\sigma}$$

Here μ and σ refer to the average and standard deviation of all points x_i. Data with the absolute Z score bigger than 3 is considered as outliers.

Apart from using Z score test to remove the outliers, we also implement the normalization process using *sklearn.preprocessing.MinMaxScaler* libaray to map all the values to the range from 0 to 1, for a better training process.

2.2.5 Experiment

In this experiment, we make different attempts to predict the threshold using the features derived from gray value of an image.

First, we try testing on the numbers of the features used in this experiment. There are seven features in this experiment, and the numbers we try are from 2 to 7. The reason why we do this attempt is to lower the influence of unimportant features. Besides, some features might have potential relationship and have mutual influence, since they are all achieved from the gray values. We believe the attempt of lowering the features used could decrease the multicollinearity.

Second, we use different regression methods to do the prediction. Six different regression models are used in this experiment.

Third, we use different random seeds to do each of the experiments, from the data train-test split step to the actual predicting. 10 random seeds are used in each set of combination. Mean values and standard deviations are used to evaluate the performance of the model. For the best combination we achieve, intersection-over-union (IoU) is used to see the effect of prediction model on the images.

The experiment procedure is shown in Figure 2.

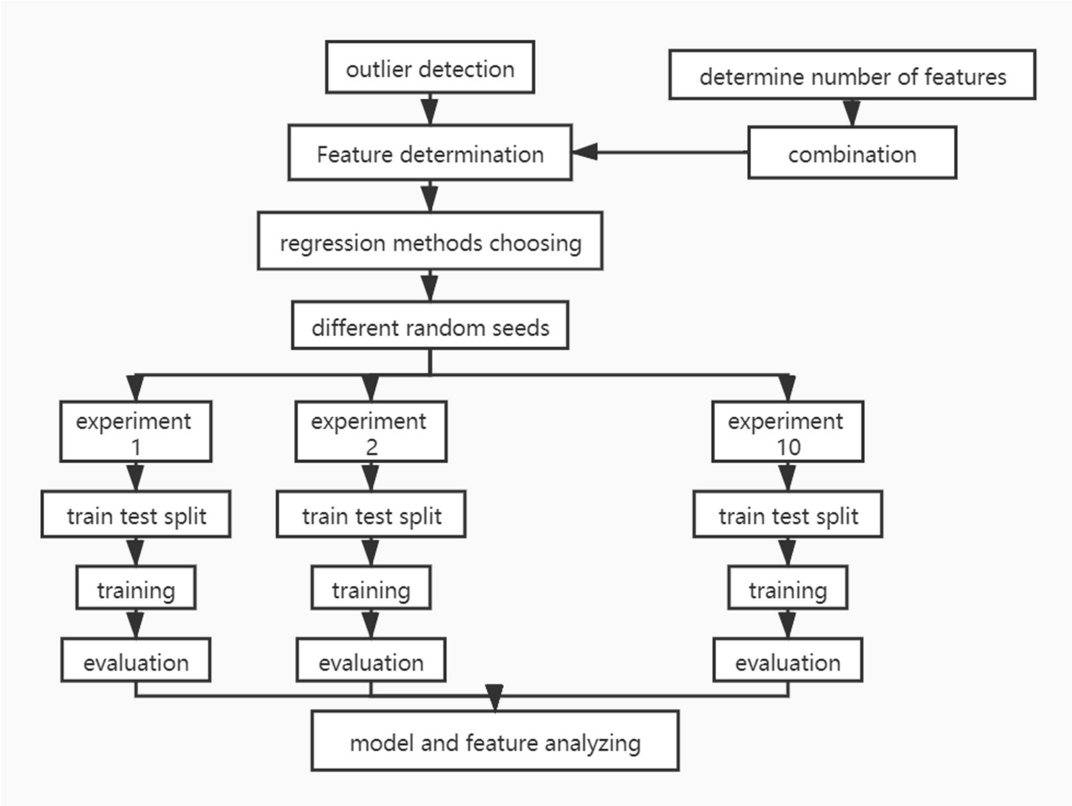


Figure 2. Experiment procedure of machine learning methods

2.3 Fuzzy C-means

2.3.1 Algorithm

Fuzzy C-means clustering is a special method of clustering algorithm. In this method, one data point could belong to 1 or more clusters, which makes its biggest difference compared to other clustering algorithms.

In Fuzzy C-means algorithm, a degree function is used to show the degree that a data point belongs to a certain cluster, and the algorithm can be expressed into the optimization problem below:

$$\text{minimize } J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left|\left|x_i - c_j\right|\right|^2$$

$$\text{s.t.} \quad \sum_{i=1}^{C} = 1, \quad j = 1,2,\dots,N$$

$$\sum_{j=1}^{C} u_{ij} = 1, \qquad u_{ij} \geq 0$$

Here u_{ij} is used to illustrate the degree that a data point *i* belongs to a class *j*. The fuzzifier *m* is used as a factor of the fuzziness, where larger *m* expresses a fuzzier cluster. When *m* equals 1 the algorithm is not fuzzy any more, in this case one data point can only belongs to one cluster. c_j is the center of cluster j. All distances in this algorithm are Euclidean distance.

2.3.2 The pseudo code of Fuzzy C-means algorithm

FCM algorithm [23]

Begin
  Initialization $c, m, T, \varepsilon, u^{(0)}$
  $\mathrm{t} \leftarrow 0$
  for t $< \mathrm{T}$
    $\mathrm{t} \leftarrow \mathrm{t} + 1$
    compute centroid $V^{(0)}$
    update $u^{(t)}$
    until $\mathrm{t} = \mathrm{T}$ or $\left\|u^{(t)} - u^{(t-1)}\right\| \leq \varepsilon$
  return $u^{(t)} - v^{(t)}$
End

2.3.3 Experiments

The implementation of FCM is based on *skfuzzy* package (https://github.com/scikit-fuzzy/scikit-fuzzy) in *Python*. Some of the experiments refer to the codes tutorial (https://github.com/ariffyasri/fuzzy-c-means).

2.4 U-Net

2.4.1 The principle of U-Net neural network

U-Net is a fully convolutional network that is well-used in image segmentation. Compared to traditional convolutional network, it has an expanding path that restores the feature map to nearly a same size as the original input image. In this case, the labels can be assigned to every pixel of the feature map, which makes it possible to do a detailed segmentation.

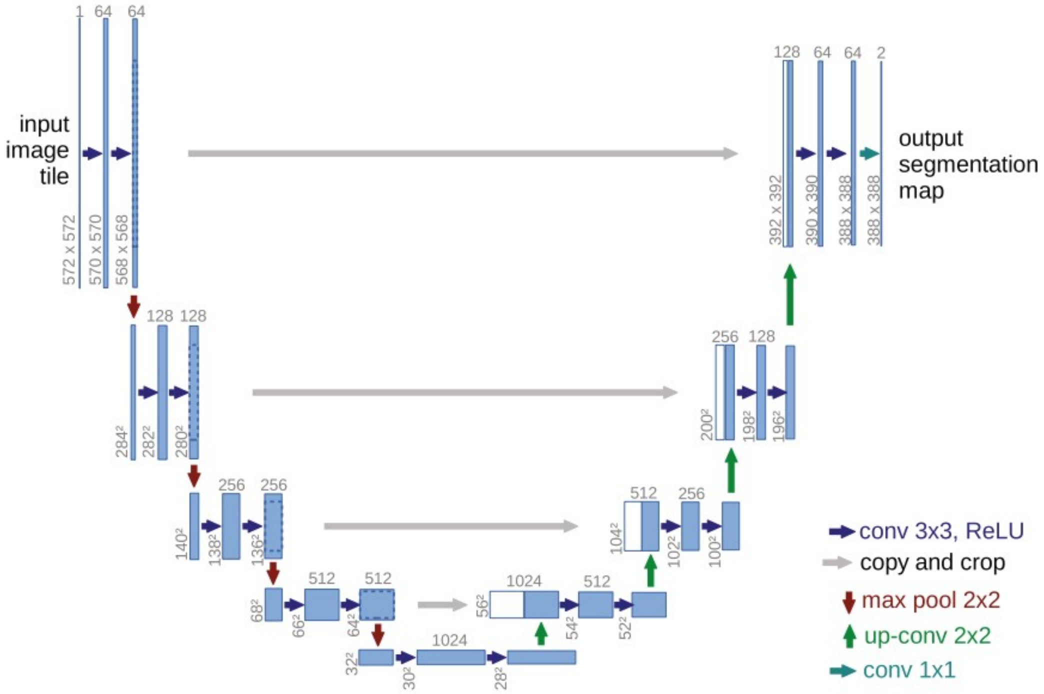


Figure 3. The structure of U-Net[21]

Figure 3 shows the basic architecture of U-Net. The contracting path and the expansive path show similarly symmetric, forming the U-shape structure. The contracting path, or the left part, is consisted of 4 blocks. Each block contains 3 convolutional layers with 3×3 kernel, and a maxpooling layer of 2×2 kernel for down-sampling. When the input size of original image is 572×572, it achieves feature maps of 32×32. On the right is the expansive path of U-Net, which is also consisted of 4 blocks. In each block, a deconvolution layer (upsampling) firstly doubles the size of the feature map. Then a concatenation layer is used to connect the correspondingly feature map of contracting path. Then two convolutional processes with 3×3 kernels are added. After the four blocks, it returns a feature map of size 388×388, which is relatively close to the size of the original image[21].

One advantage of U-Net is that it does not require a large amount of original data, but it requires data augmentation to bring the invariance and robustness[21]. Thus, data augmentation is very important in this case. According to the work[21] of Ronneberger et al., they only use 30 images of 512×512 size. For image data, commonly used augmentation methods are shifting, rotating, and mirroring processes.

2.4.2 Data pre-processing and train-test set chosen

The original size of an image is 1328×1048. We crop the original image so as to match the size of an input image of the U-Net, and we use different size for the training. We make three attempts on the size of image: large size (1024×1024), middle size (512×512), and small size (256×256).

Since the size of the whole dataset is relatively small (only 91 images in total), the data augmentation step is very essential for the long period of training. Here we use *ImageDataGenerator* from *tf.keras* library. In our experiment, an image could be randomly rotated, shifted or flipped. However, the lightness is not changed, and rescaling is not used either so as to keep the resolution of the original image. Briefly, the image could only be shifted, rotated, or mirrored, but the shape and scale keep the same. Besides, the filling mode is set to 'mirrored', which is consistent with the principle of U-Net, in which all boundaries of an image are mirrored to increase the accuracy. In each of the augmentation fold, the whole image set is augmented 5 times, which makes 455 (5×91) images of large size (1024×1024) images, 1820 (5×91×4) images of middle size (512×512) images, or 7280 (5×91×16) images of small size (256×256) images.

2.4.3 Network initialization

We use different tensor channels (the parameter *filter* in *Conv2D*) for different image size. Since our implementation of U-Net is based on Google Colaboratory, in all experiments we try to keep the channels to its maximum, for a better training result. For big size images, the channels are set to 5, while in middle and small sized images, the channels are set to 10 and 16 respectively.

2.4.4 Training and evaluation

Our strategy is using a whole augmentation fold as the test set. As a balance of training effect and calculation limit, we use one augmentation fold to do enough epochs of training, then save the training record, and use another augmentation fold to continue the training. At last, we use a new augmentation fold as the test set, and do the following evaluation. The evaluation is based on IoU and its distribution.

Some codes that are used in this experiment refer to the U-Net codes tutorial on GitHub (https://github.com/decouples/Unet/blob/master/unet.py). The experiment procedure is shown in Figure 4.

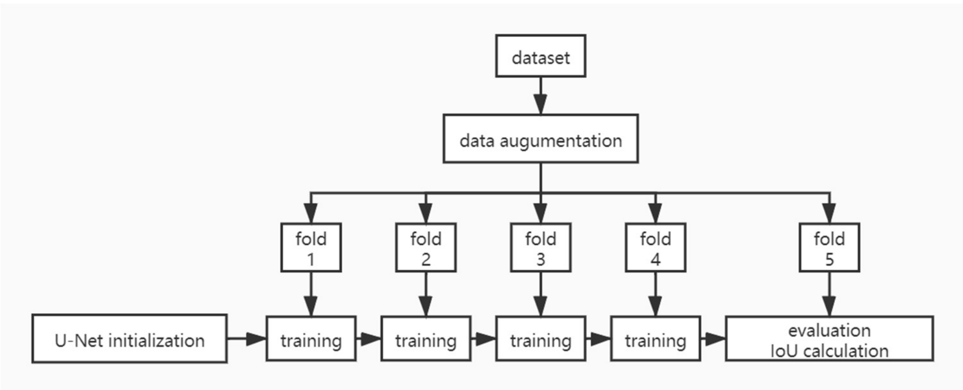


Figure 4. Experiment procedure of U-Net

Chapter 3 Results

3.1 Threshold prediction using machine learning methods

In this task, our aim is to predict the threshold of an image based on 7 features of gray values of an image. We use 6 different regression models in the prediction. We also accomplish two different attempts. First, we test on the number of features used in the prediction, from 2 to 7. Second, we also test on different combinations of the features.

In this experiment, we use two index to find the best combination of the feature set. The first is the root mean square error (RMSE) between the true thresholds and predicted thresholds. We use RMSE to evaluate the accuracy of the prediction model. The second is the standard deviation of the predicted value. It is used to check whether the prediction could keep stable. Besides, for the best combination of features, we use intersection-over-union (IoU) to see its performance on given images.

Also, we use different random seeds to do multiple experiments, for every machine learning algorithm and every feature combination. The experiment with the least RMSE among 10 tests (in different random seeds) is chosen. The results are shown in Table 2. The best combination we achieve in this experiment (thickened and italicized in Table 2) is using polynomial regression model based on 2 features: StdDev and Median. The minimal RMSE of 10 tests are 8.14, and the standard deviation in 10 tests are 3.11. We use this model on the dataset images to see the model's actual effect on images. The result is poor, with an average IoU of only 0.56. Furthermore, we calculated the best combination of features in each experiment, however no obvious feature bias is found. The detailed result can be seen in supplemental materials (Table S1). Since none of the feature is shown very helpful in threshold prediction, we do not regard this experiment as a successful one. We further search for other promising segmentation method for this project.

	Features	Min RMSE of 10 tests	Std of 10 tests
LINEAR	2	9.11	2.85
	3	8.87	2.92
	4	8.91	3.11
	5	10.12	3.65
	6	11.51	4.02
	7	15.06	5.91
LOGISTIC	2	9.65	4.66
	3	9.93	4.40
	4	9.88	4.30
	5	10.46	4.25
	6	10.76	4.43

	7	12.31	5.02
RIDGE	2	9.09	3.02
	3	8.92	3.03
	4	9.18	3.18
	5	9.82	3.18
	6	10.16	3.38
	7	11.06	3.90
LASSO	2	9.17	2.87
	3	9.01	2.90
	4	8.99	3.18
	5	10.02	3.55
	6	11.04	3.76
	7	13.63	5.42
POLY	***2***	***8.14***	***3.11***
	3	8.50	3.22
	4	10.02	3.30
	5	14.59	5.07
	6	21.40	3.29
	7	37.23	16.80
BAYESIAN	2	9.11	2.91
	3	9.04	2.90
	4	9.01	3.18
	5	9.87	3.42
	6	11.06	3.70
	7	14.43	5.66

Table 2. Minimal RMSE and standard deviation of ten tests in threshold prediction.
(Best result is thickened and italicized)

3.2 Fuzzy C-means algorithm

Fuzzy C-means algorithm is our second attempt. Since cells are not uniformly scattered in every part of the image, and the fluorescent signal is not evenly distributed, we try to use local thresholding method in the very beginning. We try splitting the image into 16 grids, and do the segmentation inside each grid. However, the result is far from satisfying, since the segmented images show very obvious severance in the border of grids. Some sample results of gridding before segmentation are shown in Figure 5. Therefore, we do not use gridding in the following experiments.

Some successful results of Fuzzy C-means clustering segmentation are shown in Figure 6. The mean IoU in Fuzzy C-means method is 0.75. Besides, the average running time for segment a 1328×1048 sized image is 192.33 second, which is relatively a long time. What's more, FCM does not always bring a good result. It is obviously shown in Figure 7 that it fails to segment the images with too many or too few cells in vision.

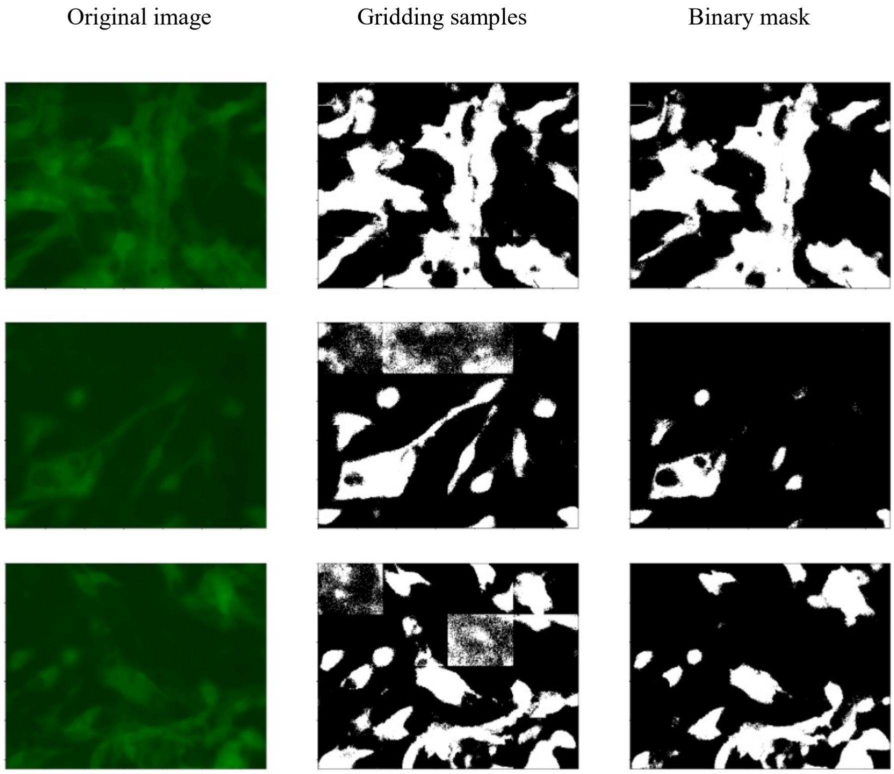


Figure 5. Sample images of gridding before implementing FCM.

Left, middle, right columns show original image, gridding samples, binary mask. Very obvious discontinuity can be seed in middle column images.

Original image	FCM segmentation (in green channel)	FCM segmentation (binary color)	Binary mask

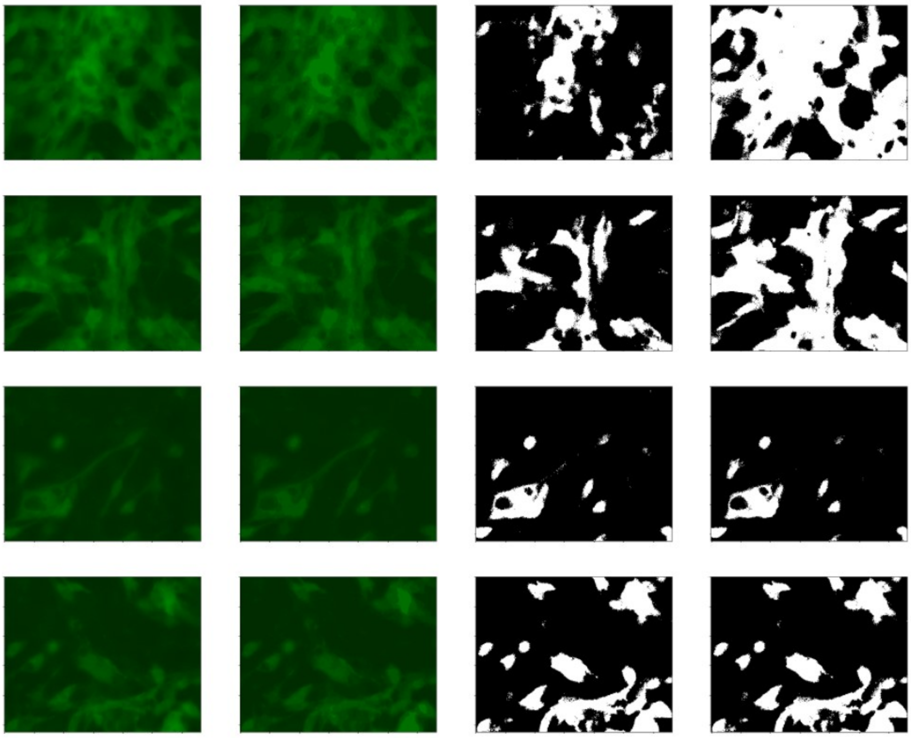


Figure 6. Sample images of successful examples using FCM.

(from left to right) First column shows the original image. Second and third columns show the segmentation result by FCM, shown in green channel image and binary color image. Fourth column shows the binary mask.

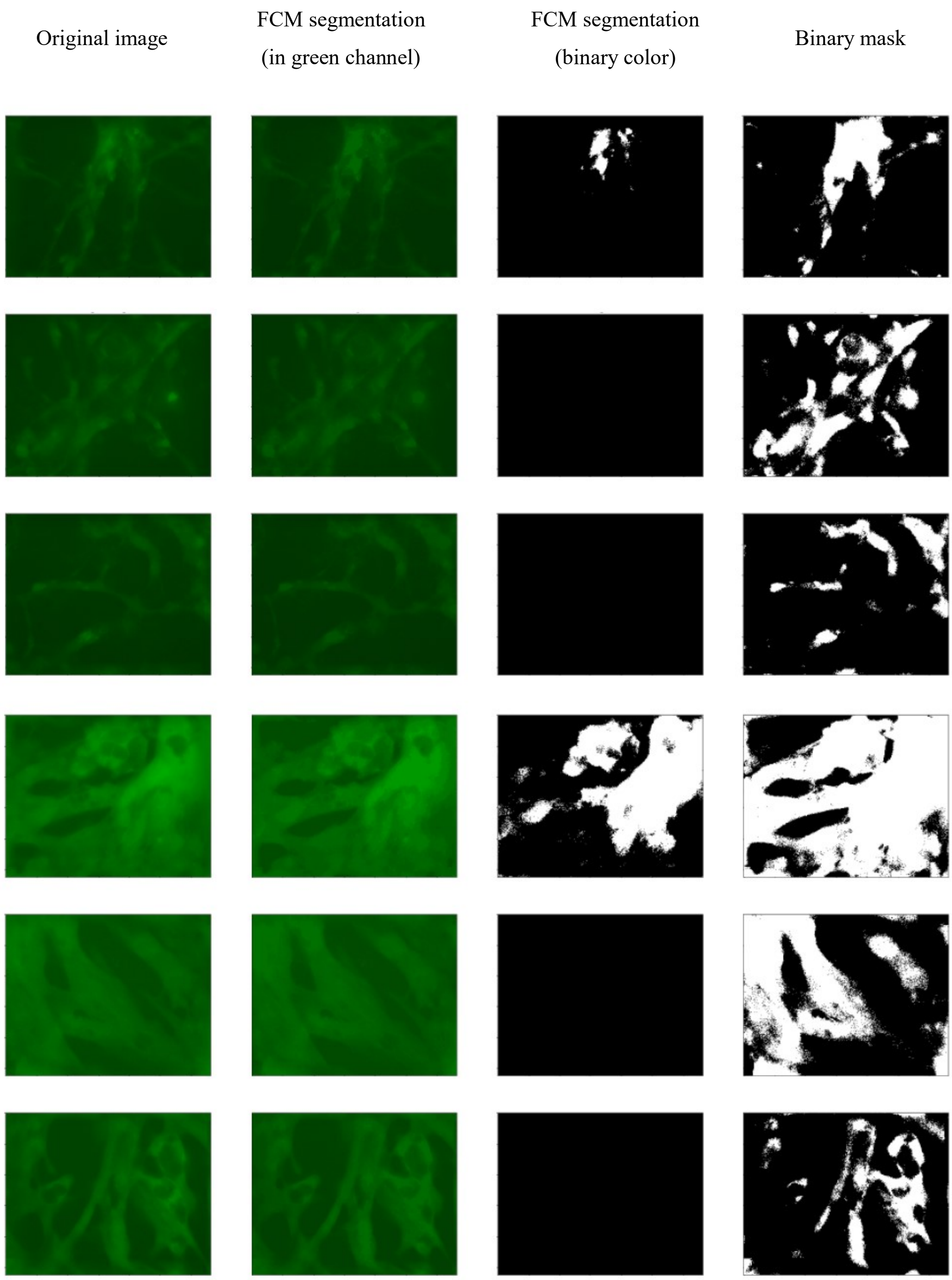


Figure 7. FCM fails to segment florescent photos with poor shooting, with too many or too few cells in vision.

3.3 Segmentation using U-net

We firstly have a quality check of the augmented images to see if the images can be used for further training. Some augmented images are shown in Figure S1 in supplementary materials. Most of the images are correctly augmented and can be used in training.

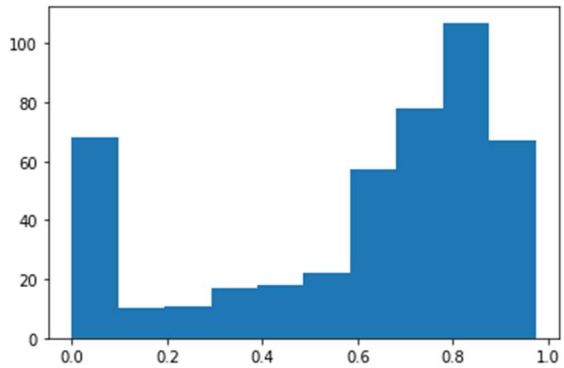
The training results of segmentation with different input image size are shown in Table 3. In the experiments with small size images, a serious overfitting can be easily found from the binary accuracy of training and test sets.

Image size	Small (256×256)	Mid (512×512)	Big (1024×1024)
Folds used in training	4	4	4
Binary accuracy (training set)	0.8083	0.8052	0.7909
Binary accuracy (test set)	0.6421	0.7382	0.7556
IoU	0.6011	0.7579	0.8468

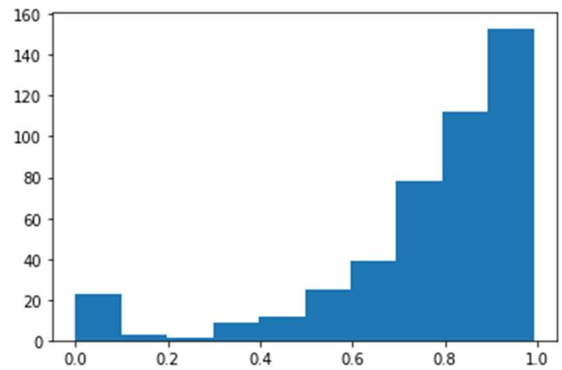
Table 3. Results of U-Net training in three different sized image

Furthermore, we calculate its final IoU distribution, to see if most of the segmented images have good IoU. The IoU distributions of experiments on three sized images are shown in Figure 8. It is very obvious that the experiment with big size images have the best performance, with higher IoU, compared with the experiments of other two sizes. Besides, we also draw the plots of training record curve, which are shown in Figure S2 in supplementary material.

Small (256×256)



Mid (512×512)



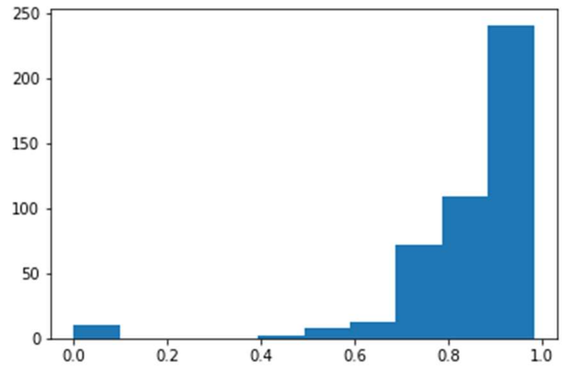
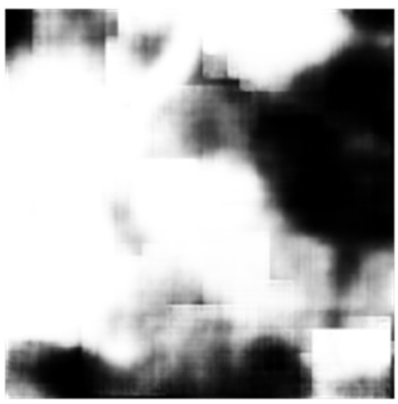
Big (1024×1024)	

Figure 8. IoU distributions of experiments on small, mid, and big size images.

In order to check the different segmentation quality of U-Net with different input image size, we randomly select some augmented image, as well as their segmentation results. The results are shown in Figure 9. In the experiment with small size image (256×256), although the segmented image shares some cell outline with the binary mask, the difference between them is very obvious. The segmentation quality becomes better in mid-size experiment. In big size experiment, the segmentation quality is very good, and the boundaries of the segmented image are very smooth, which is different of the mosaic pattern in mid-size and small-size experiments.

	U-Net segmented image	Binary mask
Small (256×256)		
		

Mid
(512×512)

Big
(1024×1024)

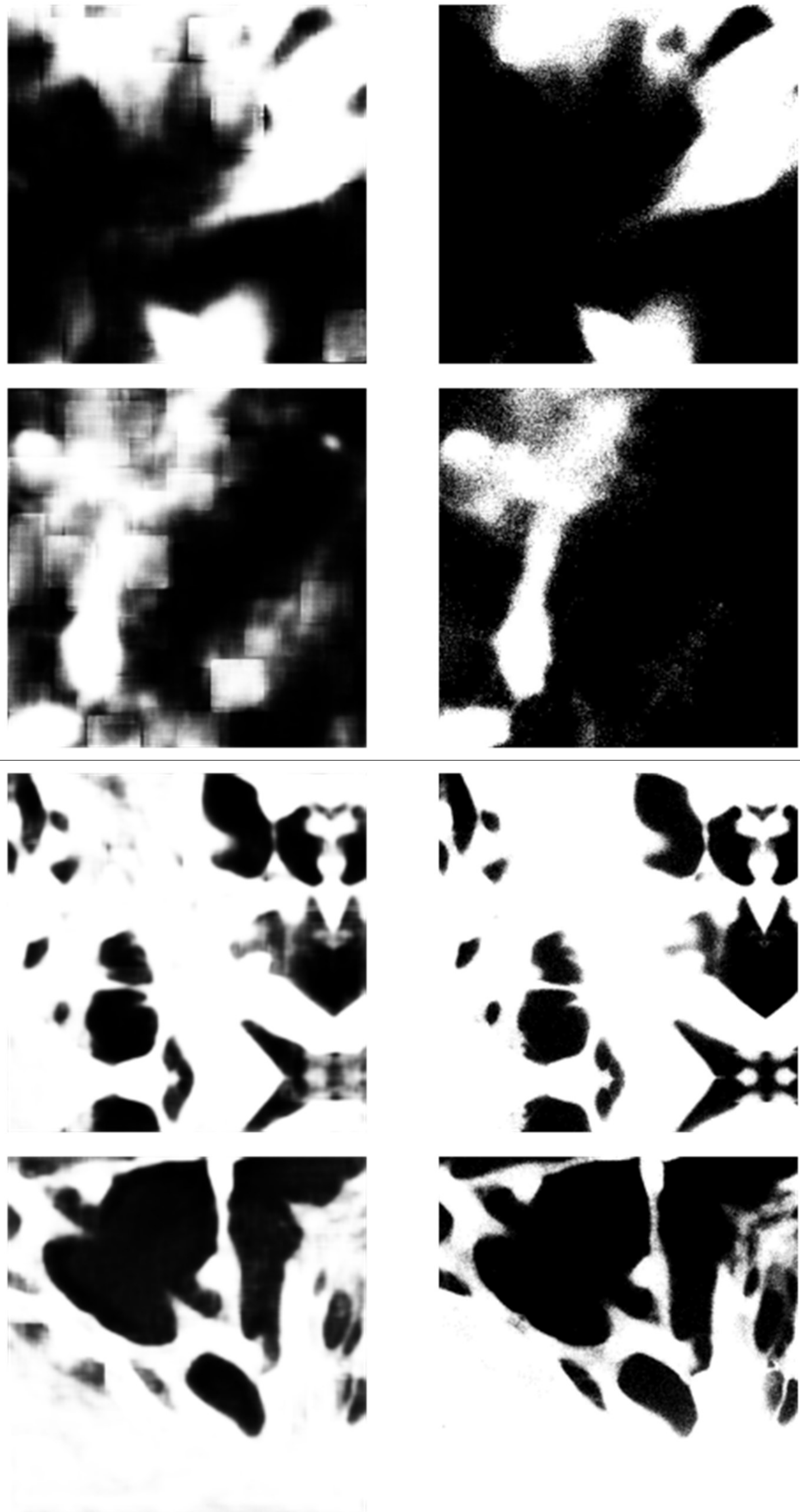


Figure 9. Some samples of segmentation result of U-Net with different input image size

Chapter 4 Discussion

4.1 Threshold predicting using machine learning method

For machine learning method, the quality control of the original images is a very important step since outliers of the dataset make the model less accurate. In this experiment, we do the outlier detection according to their gray value features, because florescent images with abnormal gray value distribution could indicate that the image does not contain a normal cell ratio, or even only shows the background. We detect seven outliers in total. In addition, we check the corresponding images, and find that these are indeed florescent images, with no obvious cell in the field. Thus we believe that the pre-processing of detecting the outliers is successful. However, it might suggest that the predicting of machine learning methods cannot handle the outliers very well, and requires additional step for a better result.

However, it suggests that the threshold prediction based on gray value features does not achieve a satisfying result. In the best combination of features, the minimal RMSE between true and predict value is 8.14. Since the images in the dataset are 8-bit, which means that the gray value range of a pixel is from 0 to 255, a RMSE of 8.14 is actually a big difference. An IoU of only 0.56 also confirms the conclusion. Besides, the standard deviation is 3.11, which means that the effect of the threshold prediction is not stable, but with a large fluctuation. Also, since no feature bias is shown in the experiment, we cannot distinguish those important features from the unimportant ones, which enhance the underfitting in the training processes. With all the disadvantages discussed above, we then have the following experiments with other methods.

4.2 Fuzzy C-means algorithm

As an unsupervised learning algorithms, FCM has a good performance in this fluorescent image segmentation problem. Usually, fluorescent images are taken by human, with carefully chosen vision and finely tuned image. However, in this task, images are achieved by automatic workflow, which brings a small part of images with poor vision, having too many or too few cell in the vision field. FCM fails to handle these images. But for those well-taken images, FCM has a good performance.

Another problem is that the task of distinguishing the background and the foreground is actually a 2-cluster problem. In actual experiment, we try to increase the number of clusters, and obtain some interesting results (shown in Figure S3 in supplemental material). With bigger number of clusters, the segmented images tend to identify more patterns and textures of the original images. One reasonable assumption is that we could choose from image outputs of different cluster numbers. However, when the number of clusters reaches 7, the running time of a single image (1328×1048) comes to over ten minutes. It seems that the time consuming is not well worth trying under the existing computing conditions.

4.3 Segmentation using U-Net

Segmentation with U-Net is our final attempt. In this task, we firstly do the cropping to all original images, from 1328×1048 to 1024×1024. This step is in convenience of network size design and better comparing among experiments on different sizes. Since we use Google Colaboratory to implement the computing, we cannot use the network with bigger filters, which could be a disadvantage. In actual experiment, we try some filter value that is less than what we mentioned in previous chapter. It shows that if the value of filter is not big enough, the network cannot reach the convergence, and the segmentation task cannot be finished in this case. Also, since the experiment of small size images (256×256) shows very poor performance, we conclude that our segmentation task is not successfully done in this size.

Another attempt in U-Net we try is the mathematical morphology. In Figure S1 we can see obvious pixel particles alone from the cell monolayer. We try opening and closing operations in order to remove them. However, the performance is not satisfying. Besides, the opening and closing operations are really time consuming steps, because the images are already augmented. That’s why we do not add them to the following experiment.

4.4 Other

One limit of the segmentation task is the binary mask it uses. As we mentioned before, those binary masks are achieved from manually set thresholding by biologists. However, one disadvantage is that this simple thresholding method bring a lot of noises (the pixel particles) in the mask, which makes the training performance less accuracy. For further work, binary masks with finely circled foreground should take the place of the current ones. However, this can be regarded as a challenge for further work, since finely labelled dataset is always difficult to create.

Chapter 5 Supplemental Materials

FEATURES		RMSE	Mean	StdDev	Mode	IntDen	Median	Skew	Kurt
LINEAR	2	9.11				O	O		
	3	8.87					O	O	O
	4	8.91	O	O				O	O
	5	10.12	O			O	O	O	O
	6	11.51	O	O		O	O	O	O
RIDGE	2	9.09	O	O					
	3	8.92	O	O				O	
	4	9.18		O	O			O	O
	5	9.82	O	O		O	O		O
	6	10.16	O	O		O	O	O	O
LASSO	2	9.17		O			O		
	3	9.01				O	O	O	
	4	8.99		O	O			O	O
	5	10.02	O			O	O	O	O
	6	11.04	O		O	O	O	O	O
POLY	***2***	***8.14***		***O***			***O***		
	3	8.50	O	O		O			
	4	10.02	O		O	O	O		
	5	14.59	O	O	O		O		O
	6	21.40	O	O	O	O	O		O
BAYESIAN	2	9.11				O			O
	3	9.04	O					O	O
	4	9.01	O		O			O	O
	5	9.87	O			O	O	O	O
	6	11.06	O		O	O	O	O	O

Table S1. Detailed features chosen and RMSE in threshold prediction using regression models.

Symbol O indicates that it is one of the features in the best combination.

Results of predicting based on all 7 features are not shown.

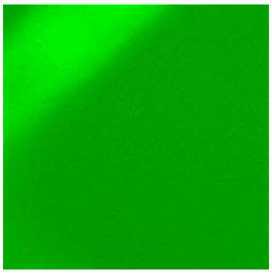
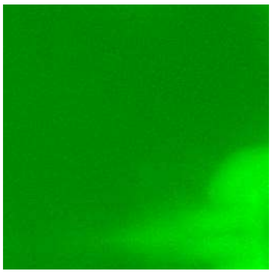
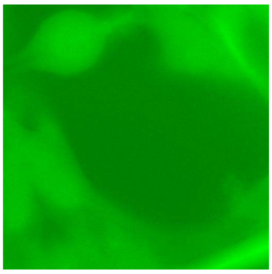
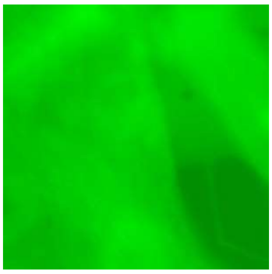
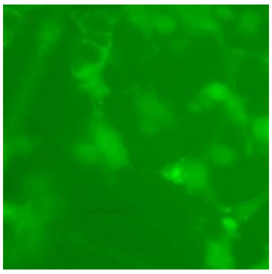
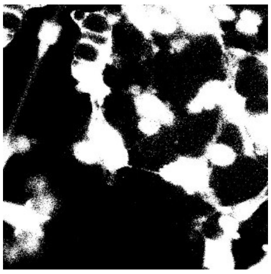
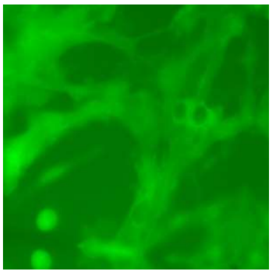
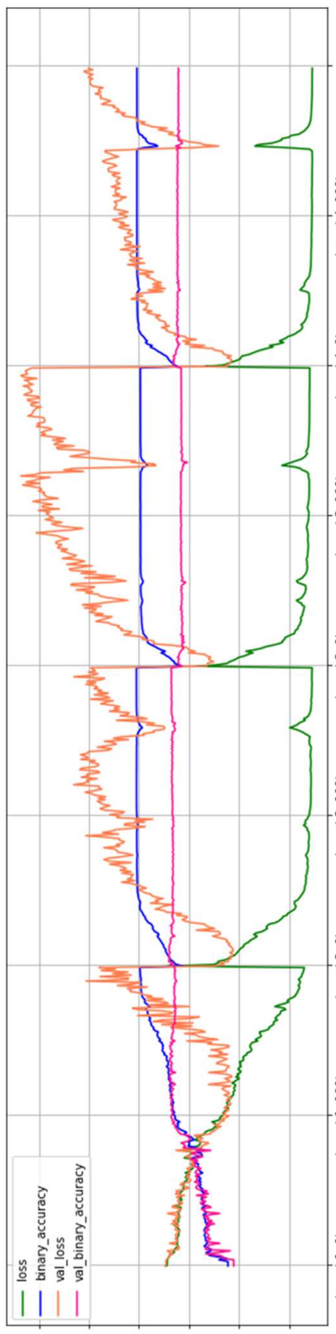
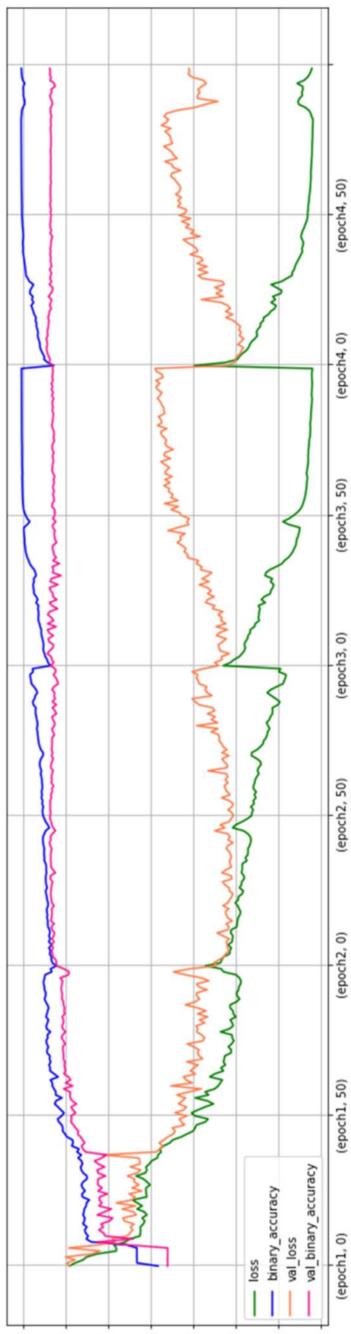
Image size	Image	Binary mask
Small (256×256)		
		
Mid (512×512)		
		
Big (1024×1024)		
		

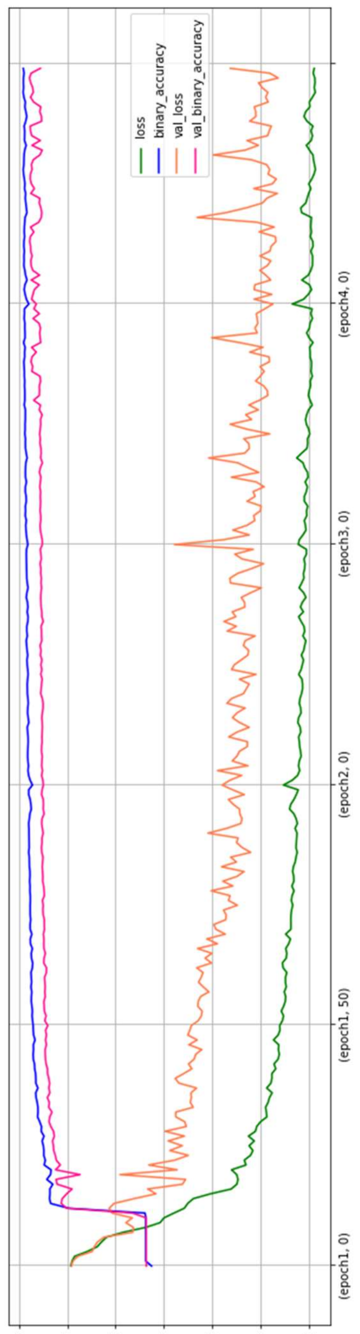
Figure S1. Some samples of augmented image data.



Small size (256×256)



Mid size (512×512)



Big size (1024×1024)

Figure S2. Training curve of three sized images

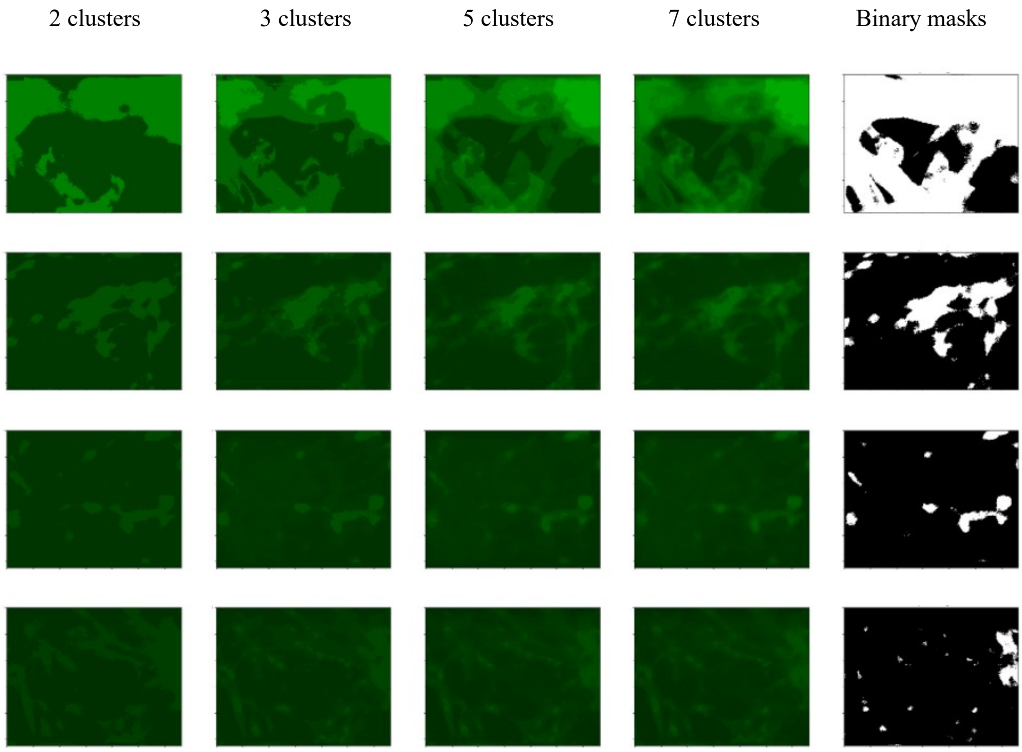


Figure S3. FCM using different cluster numbers.

(From left to right) the number of cluster = 2, 3, 5, 7. The images of the right column is the binary mask.

Reference

[1] P.W. Burridge, Y.F. Li, E. Matsa, H. Wu, S.-G. Ong, A. Sharma, A. Holmström, A.C. Chang, M.J. Coronado, A.D. Ebert, Human induced pluripotent stem cell–derived cardiomyocytes recapitulate the predilection of breast cancer patients to doxorubicin-induced cardiotoxicity, Nature medicine 22(5) (2016) 547-556.
[2] M.C. Ribeiro, R.H. Slaats, V. Schwach, J.M. Rivera-Arbelaez, L.G. Tertoolen, B.J. Van Meer, R. Molenaar, C.L. Mummery, M.M. Claessens, R. Passier, A cardiomyocyte show of force: A fluorescent alpha-actinin reporter line sheds light on human cardiomyocyte contractility versus substrate stiffness, Journal of molecular and cellular cardiology 141 (2020) 54-64.
[3] T. Ferreira, W. Rasband, ImageJ user guide, ImageJ/Fiji 1 (2012) 155-161.
[4] T. Ridler, S. Calvard, Picture thresholding using an iterative selection method, IEEE trans syst Man Cybern 8(8) (1978) 630-632.
[5] J.M. Prewitt, M.L. Mendelsohn, The analysis of cell images, Annals of the New York Academy of Sciences 128(3) (1966) 1035-1053.
[6] N. Otsu, A threshold selection method from gray-level histograms, IEEE transactions on systems, man, and cybernetics 9(1) (1979) 62-66.
[7] R. Adams, L. Bischof, Seeded region growing, IEEE Transactions on pattern analysis and machine intelligence 16(6) (1994) 641-647.
[8] L. Najman, M. Schmitt, Watershed of a continuous function, Signal Processing 38(1) (1994) 99-112.
[9] D. Ziou, S. Tabbone, Edge detection techniques-an overview, Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii 8 (1998) 537-559.
[10] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained k-means clustering with background knowledge, Icml, 2001, pp. 577-584.
[11] A.A. Salihah, M.Y. Mashor, N.H. Harun, H. Rosline, Colour image enhancement techniques for acute leukaemia blood cell morphological features, 2010 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2010, pp. 3677-3682.
[12] K.M. Bataineh, M. Naji, M. Saqer, A Comparison Study between Various Fuzzy Clustering Algorithms, Jordan Journal of Mechanical & Industrial Engineering 5(4) (2011).
[13] M. Song, D. Civco, Road extraction using SVM and image segmentation, Photogrammetric Engineering & Remote Sensing 70(12) (2004) 1365-1371.
[14] S. Wang, W. Zhu, Z.-P. Liang, Shape deformation: SVM regression and application to medical image segmentation, Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, IEEE, 2001, pp. 209-216.
[15] S. Minaee, Y.Y. Boykov, F. Porikli, A.J. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).
[16] K. O'Shea, R. Nash, An introduction to convolutional neural networks, arXiv

preprint arXiv:1511.08458 (2015).
[17] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097-1105.
[18] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[19] G. Takos, A survey on deep learning methods for semantic image segmentation in real-time, arXiv preprint arXiv:2009.12942 (2020).
[20] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431-3440.
[21] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234-241.
[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.
[23] T. Munandar, A. Musdholifah, Comparative Study Between Primitive Operation Complexity Against Running Time Application On Clustering Algorithm, International Journal of Advanced Research in Computer Science 5(5) (2014).