MATH 4432 Project 3 Report

Anonymous Author(s)
Affiliation
Address
email

Abstract

This project studies the segmentation of cells in images from fluorescence microscope using a combination of both unsupervised learning and supervised learning. To start with, we will introduce some preliminaries of the research such as the background and how the experiemnts were conducted to obtain the cell images. Then we conduct unsupervised learning using a Cut and Shrink algorithm to automatically segment cells from images and the segmentation results together with five features of pixel are used to form the training set. After that, we will discuss the usage of this automatically generated traning set: instead of setting training set as ground truth for supervised learning, we regard training set as a roughly correct result with errors. By training Support Vector Machine (SVM) with this roughly correct training set, we manage to correct some erros made by unsupervised learning, which leads to even more accurate segmentation of cells. The result suggests that with properly selected features and model, unsupervised learning may have the potential to replace the tedious manual labelling process for training set generation and based on the training set from unsupervised learning, supervised learning model can be trained to produce even better results.

1 Introduction

1.1 Background

Machine learning has been one of the most heated methods that are being applied to revolutionize the researches in biology, in which one of the most basic and essentail steps is to segment cells in images from microscope for further studies on the segmentation results. When viewed from the perspective of machine learning, cell segmentation can be treated as a classifcation problem where each pixel in the image needs to be classfied as either cell or background. Although supervised learning can genearte powerful models to perform cell segmentation, it would be time-consuming to manually identify each pixel of cell images by eyes so as to produce a training set for supervised learning. Furthermore, in reality, due to technical problems, the input images may have defects such as non-uniform illumination and noisy background, which renders automatic cell segmentation with high accuracy even more challenging. This motivates us to come up with a method where unsupervised learning and supervised learning are combined together to fully automate the cell segmentation process. More specifically, unsupervised learning is first used to segment cells and the results will be used as training set for supervised learning model. The core idea is that due to the challenging nature of cell segmentation for some images with poor quality, unsupervised learning may make some mistakes and we may rely on supervised learning to correct these mistakes if the model can learn with 'critical thinking' by not fully fitting the results from training set.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

1.2 Data Description

For data, we use in total five cell images of size of 512×511 pixels as raw data to be segmented by unsupervised learning. The training set is generated by trainsforming the images into .csv files with the first and the second columns to be the 2D coordinate of each pixel in each cell image. The third column in the .csv file contains the scaled intensity of each pixel where the intensity value of each pixel is divided by the smallest intensity value in the image. The fourth one contains the magnitude of the gradient of each pixel from the scaled image. The fifth column is the vincinity information of a pixel, which we define as the average of the intensity of the surrounding eight pixels(for pixels on boundaries, we replicate the images such that they can have eight surrounding pixels). And finally, the sixth column contains the indicator whether the pixel belongs to a cell (shown as 1) or the background (shown as 0) based on results from unsupervised learning.

Moreover, the first image will serve as a training set and the second image will be the validation set for model optimization. Note that we are comparing the unsupervised method with the supervised method as well as comparing results from both methods with the ground truth (the raw images). Thus, the first two images are generally regarded as the training set that helps us find the best model and the interpretation of why the model outperforms others. Finally, the last three images will be used as the testing set that evaluate the performance of the best model. These three images are chosen to challenge the optimized supervised learning model because when compared with training set, they have some challenging features such as high background noise or cells with low fluorescence intensity.

2 Overview of Methodologies

Live cell imaging with fluorescence microscope is utilized to generate images of HeLa cell expressing green fluorescent protein (GFP). The experimental setup will be briefly introduced in part 3.

The so-called Cut and Shrink unsupervised learning is conducted by using a combination of marker controlled watershed algorithm and two component Gaussian mixture model, which will be discussed in detail in part 4. The script for unsupervised learning is written in Matlab due to its powerful image processing toolbox.

Supervised learning is conducted using SVM. The reason for choosing SVM and its performance will be discussed in part 5.

3 Experiments to Obtain Cell Images

Cells in all the images are from a HeLa cell line engineered to express GFP. Imaging was performed on a fluorescence microscope with on-stage incubator to maintain a cell-friendly environment during imaging. GFP expressed by cells was excited by light at 475 nm and the emitted fluorescence at 509 nm was recorded by a CMOS camera to produce the image.

4 Training Data Generation with Unsupervised Learning

4.1 Some Preliminaries

To clarify the idea and the essential purpose of this step, it should be reiterated that we are using the outcomes from the unsupervised learning as the labels of the training data.

4.2 Cut and Shrink Algorithm

The image to be segmented (Figure 1) is first cut into different regions containing exactly one cell using a marker-controlled watershed algorithm. Briefly, in this marker-controlled watershed algorithm, markers of cells are first identified. As can be seen from Figure 2, markers (regions highlighted in white) are regions which can be judged to be cells with very high confidence due to its relatively high intensity, but at the same time, regions identified as markers are much smaller than the whole cell. Therefore, for multiple overlapped cells, their markers do not overlap in most cases and

thus markers can effectively help avoid mistaking multiple overlapped cells as one cell. After that, the image is roughly cut into multiple regions such that each region contains one and exactly one marker with normal watershed algorithm which identifies boundary based on gradient profile of the image (Figure 3).

Figure 1: Raw Image

Figure 2: Image with Markers

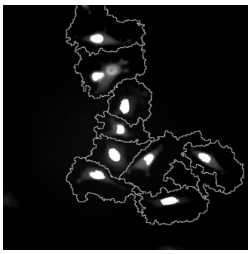

Figure 3: Boundaries by Watershed Algorithm

At the stage of Figure 3, we have already completed the 'Cut' part of the algorithm. Next we need to 'Shrink' the areas cut out in the previous step to further narrow segments of the cells down to more accurate ranges. We have utilized a two component Guassian mixture model, which has to make the following assumptions: the intensity of each pixel in each region cut out in the first step can be classified into two Gaussian distributions: one for fluorescence intensity of background and the other for fluorescence intensity of cell. This by itself explains why we need to cut the image into regions with only one cell as the first step. Without the Cut step, we may need to consider each cell in the images as a uniqe Gaussian distribution and this can lead to a complicated N components Gaussian mixture model where the estimation of N can also be challenging.The two Guassian distributions are independently parameterized. We are trying to infer the parameters in these two Guassian distributions iteratively to maximize the likelihood that each pixel belongs to certain cluster and stop after convergence of parameters. Based on the inferred paremeter, we classify each pixel according to the larger probability that it belongs to a certain cell or background. And the final results are the outputs of the unsupervised Cut and Shrink algorithm. For the purpose of easy visualization, we transform the result which should be a vector of 0 or 1 into a binary image.

Figure 4: Segmentation after Cut and Shrink

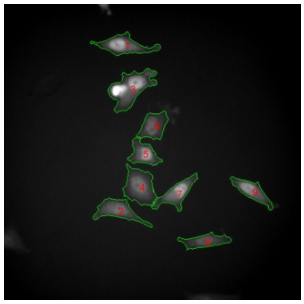

Figure 5: Error in Classification

As can be seen from Figure 4 and 5, at first glance, the Cut and Shrink method can give impressive segmentation results as all the cells in the image have been segmented with high fidelity in their shapes. The most significant advantage of this Cut and Shrink algorithm is that after optimizing some hyperparameters with one image, it can automatically label all the rest images with stable performance and thus there is no need to manually label any cell. However, the segmentation may still make mistakes if part of the cell has quite low fluorescence intensity (as shwon in Figure 5). Part of cell 3 and cell 6 has been misclassified as the background due to its low intensity. The reason can be that when handling pixels with suspicious intensity which can either be classified as background or cell, the current Cut and Shrink method only takes intensity into consideration while human eyes and brain will consider the location of pixels in addition to intensity. For instance, the Cut and Shrink method may regard suspicious pixels as background (or cell) simply because the probability for it to

come from the background (or cell) distribution is slightly higher while human eyes and brain will judge this pixel based on whether the pixels around it are from cell or background.

Therefore, future improvement can be made to include the spatial information of pixels into cell segmentation to achieve higher accuracy. This inspires us to use supervised learning with some propelly engineered features to classify each pixel in more detailed manner.

5 Supervised Learning by Support Vector Machine

5.1 Feature Selection Criterion

The training set in total conatins five columns for four different feartues: 2D coordinate of each pixel (row and column in the image matrix), scaled intensity, magnitude of gradient and mean of surrounding pixels. Choosing scaled intensity as one feature does not need further explanation. Magnitude of gradient is chosen because at the boundary of cell and background, although the intensity of the pixel may be low when compared with pixels around the center of the cell, there is still a sudden jump in fluorescence intensity. Therefore, magnitude of fluorescence intensity is expected to help improve classfication of pixels near boundaries of cell and background. Mean of surrounding pixels is chosen to preserve spatial information to certain extent because as stated before, when treating suspicious pixels, huamn eyes and brain can form judgement based on whether pixels surrounding the pixel is cell or not. Finally, it may be confusing why the 2D coordinate of each pixel is chosen as one feature. The reason is that all images are subject to non-uniform illumination. In fact, pixels at four corners of the image are inherently darker than pixels in the centre of the image. This is due to the technical problem that for excitation light, its intensity gradually declines when moving from centre of the beam to the outter part. Therefore, we choose to include the 2D coordinate of each pixel to see whether non-uniform illumination may be corrected to certain extent.

5.2 Optimization Criterion

Since we are using the outputs of the Cut and Shrink agorithm, i.e. the unsupervised learning algorithm, we are not training towards or testing against the ground truth because as stated above, unsupervised learning can make misatkes. Therefore the naive usage of error rate as criteria for model optimization is not logical because the disagreement between two methods does not necessarily indicate the truthness or the error of any model. In fact, we even desire, to some content, that there will be disagreement between the unsupervised method and the supervised one because we want to improve the performance of the Cut and Shrink algorithm. Thus, instead of using error rates, we directly visualize cell segmentation results as binary images and highlight the differences to compare results from supervised learning and choose the best model.

5.3 Incentives of Using SVM

First and foremost, we want to explain why with so many models for classfication such as logistic regression, linear discriminant analysis and tree-based methods, we directly choose SVM for supervised learning. The reason is that most pixels in the image are very easy to classify because they are either very dark or very bright. Secondly, in all images, the pixels from the background dominate and thus it is likely that background pixels can dominate the loss function. The true challenging part for classification is actually the pixels at the boundary of cell and background. Therefore, we want a model that can selectively 'learn' the most challenging part instead of being 'distracted' by those easy pixels. SVM's hinge loss function may enable it to select pixels that are truly challenging as support vectors. As can be seen from Figure 6, results from SVM agrees with our intuition as support vectors indeed majorly concentrate around the boundary of cells.

5.4 SVM with Linear Kernel

We begin from trying out SVMs with linear kernels. More specifically, we have used $cost = 0.1, 1, 10, 100$ during the training and the trained model is applied to segment cells from the validation set. The results from segmenting the validation set are quite similar with no significant differences. So here in this report, only one sample result is presented for analysis.

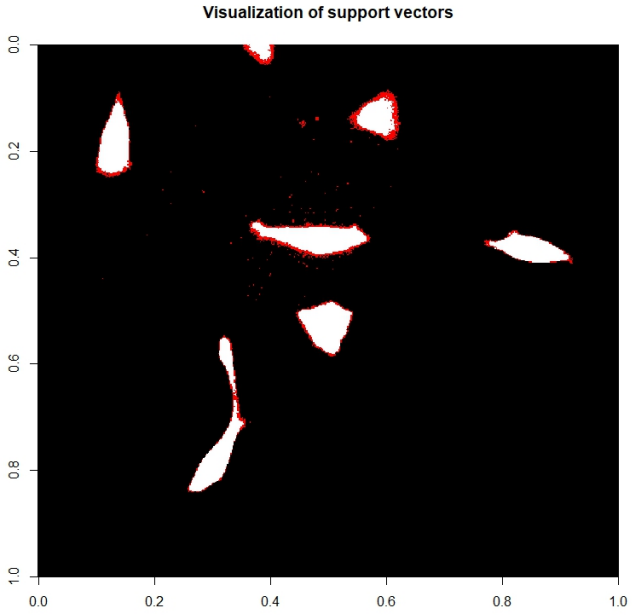

Figure 6: SVM with Linear Kernel: Cost = 10 and support vectors are highlighted in red

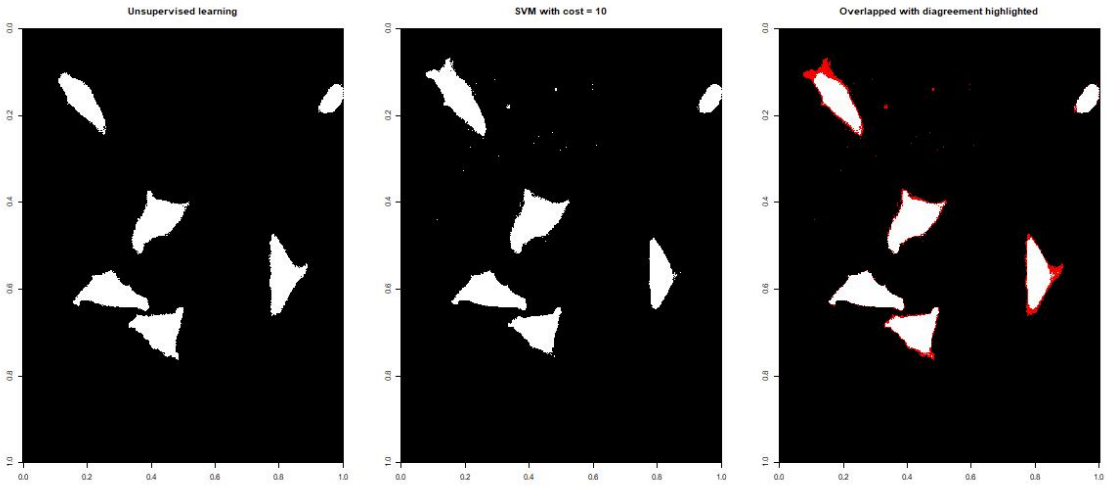

Figure 7: SVM with Linear Kernel: Cost = 10

The figure on the left of Figure 7 is the output of the unsupervised learning algorithm and the one in the middle is the results from SVM with linear kernel $cost = 10$. The graph on the right is the comparison between the former two results and the differences are highlighted in red. In fact, for the cell on the top left, the part in red is indeed part of the cell as confirmed from the raw image. So in this part, the supervised learning algorithm outperforms the unsupervised one. However, if we take a look at the cell on the middle right, the red part is actually part of the actual cell, but SVM ignores it. So, for this part, the SVM does not outperform the unsupervised learning method.

This suggests that even though the SVM has some limitation in the correctness of capturing the parts of cells, it can reveal the cell pattern that the unsupervised learning cannot. After noticing that the part where SVM performs better is close to the upper left corner of the image, we form a hypothesis that after being trained with pixels close to the corner with 2D coordinate of pixels as one feature,

SVM may indeed be able to handle non-uniform illumination to certain extent. More specifically, we think SVM may have lower threshold in intensity for pixels close to the upper left corner to be classfied as cell. The hypothesis is further supported by the fact that if we exclude 2D coordinate from the training of SVM, SVM can no longer identify the part close to the upper left corner as shown in Figure 8.

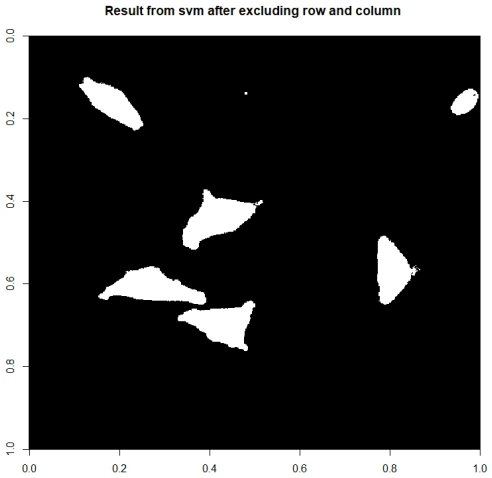

Figure 8: SVM with Rows and Columns excluded

Although SVM succeeds to correct mistakes from the unsupervised learning, we also notice that the SVM fails to capture the extra part of the cell on the middle right. Therefore, we move on to SVM with non-linear kernel to see whether a non-linear hyperplane can give even higher performance.

5.5 SVM with Non-Linear Kernel

First we try out the SVMs with polynomial kernels, the results are not very satisfying. We put the model with $Degree = 4, Cost = 10, \gamma = 0.1$ as an example. As shown in the left panel of the Figure 9. the model does capture the extra correct information of the cell on the upper left. However, it still misclassifies the areas for the middle right cell . So, we do not see much improvement from the Linear Kernel, to Polynomial Kernel.

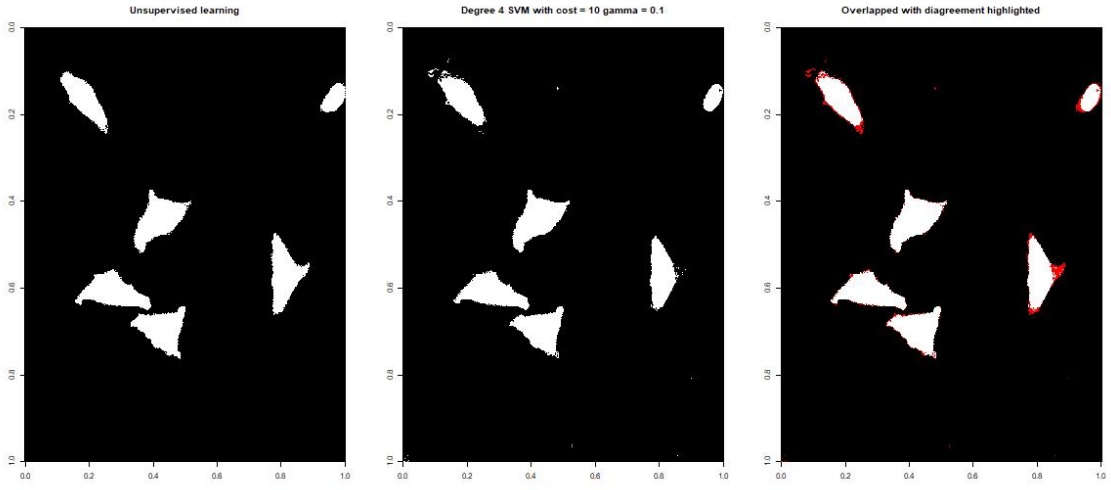

Figure 9: SVM with Polynomial Kernel: Degree=4, Cost=10, gamma=0.1

Next, we try out SVMs with radial kernels. As shown in Figure 10, the model successfully captures the extra part of the cell on the upper left as well as avoiding misclassifying the pixels around the cell on the middle. So, the SVM with radial kernel outperforms the ones with linear kernel and unsupervised learning in terms of accuracy and coorectness.

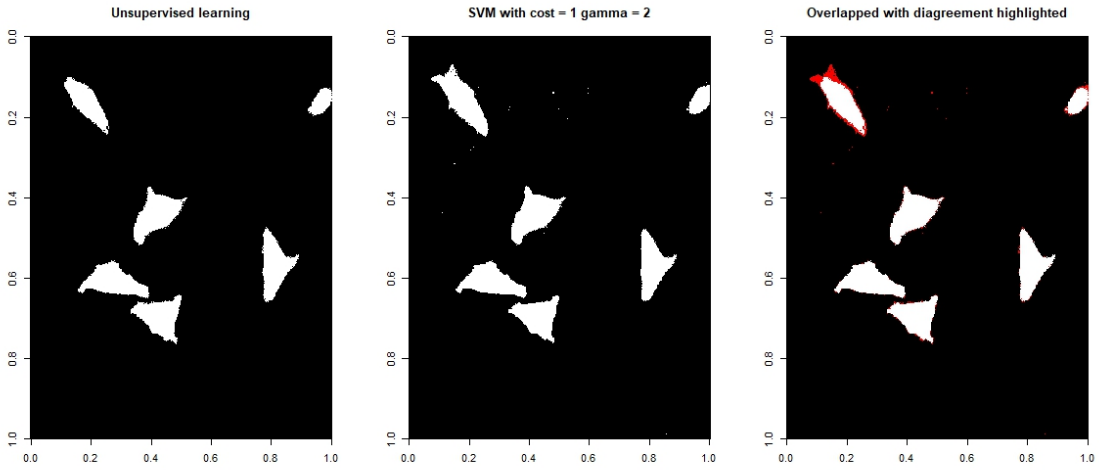

Figure 10: SVM with Radial Kernel: Cost=1, gamma=2

So, we will regard the SVM with radial kernel with $Cost = 1, \gamma = 2$ as our best model. To examine the robustness of the SVM model with radial kernel, we challenge the optimized model with three relatively challenging images.

5.6 Testings

Due to page limit of the report, we will only give a detailed analysis of one image. Another two results are attached at the end of the report for reference.

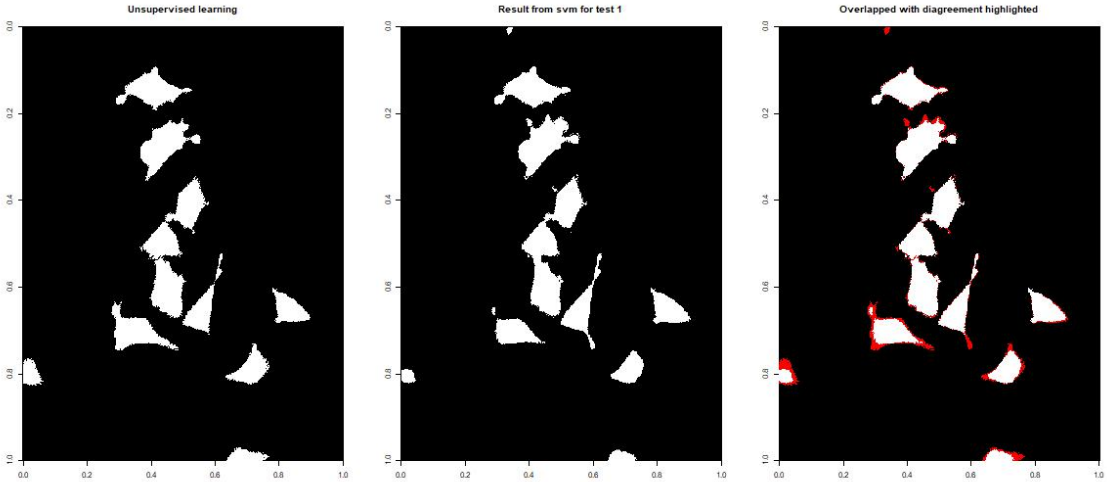

Figure 11: Testing 1

First, for clarity, the tiny objects on the boundaries of the image in Figure 11 can all be ignored because in real application, cells at the boundaries of the image will not be analysed because part of them are out of the image. Therefore, no effort has been spent on segmenting cells on the boundaries of images in either unsupervised learning or supervised learning. As can be seen from Figure 11,

SVM outperforms unsupervised learning in three cells: the second and third cell from the top and the third cell from the right. For these three cells, SVM manages to identify tentacle-like structures of cells which indeed exist in the original image. However, SVM also fails to identify some parts of the cells identified by unsupervised learning. The most significant failure is the first cell from the left whose long 'tentacle' is not identified by SVM as part of cells. Therefore, results from SVM and unsupervised learning generally agree with each other in most parts and SVM can indeed identify some subtle parts ignored by unsupervised learning, but we admit that it also makes some mistakes.

6 Conclusions

To conclude, cell segmentation in cell image processing could be challenging. The premilinary tryout of unsupervised learning method using Cut and Shrink provides us with a starting point for basic cell segmentation. The method can segment cells with acceptable accuracy. To further improve the accuracy, we have tried supervised learning method based on SVM. SVM with linear kernel have shown improvement on the details and the one with radial kernel further improves the performance of segmentation on the whole. However, the final model of $Cost = 1, \gamma = 2$ still fails to capture some of the details , which indicates a potential room to seach for other supervised classification methodology or more complex model structuring.

7 Appendix

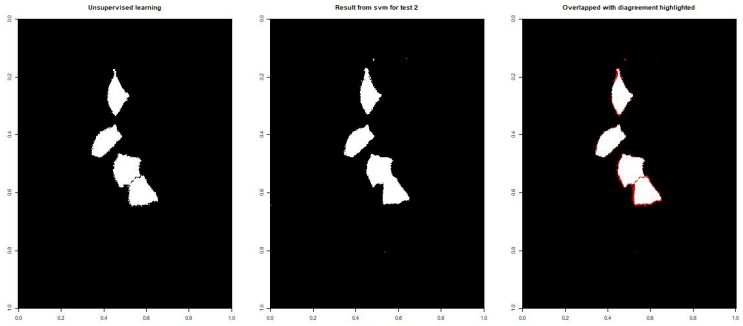

Figure 12: Testing 2

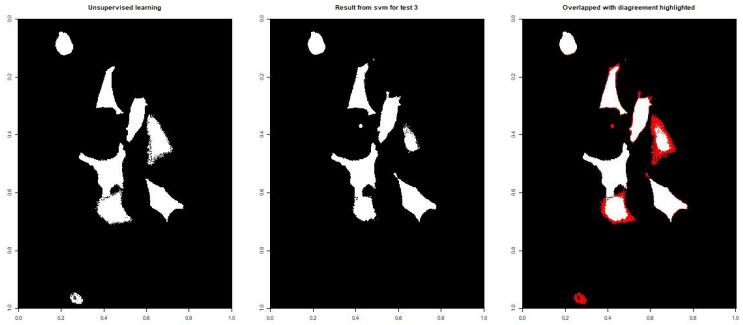

Figure 13: Testing 3

8 Contributions

Coding: Xingbo SHANG;

Project Report: Kao ZHANG.