Detection of HeLa Cells in brightfield images

Hao Peng[1], **Tao Xiang**[2], and **Zhehui Huang**[3]

[1]**Zhuhai College of Science and Technology**
1834364839@qq.com

[2]**Technical University of Munich**
tao.xiang@tum.de

[3]**Ulink College of Suzhou Industrial Park**
huangzhehui219@gmail.com

Abstract

In some occasions of cell detection and segmentation, several specific methods of separating presented touching and overlapping cell structures always need to be utilized. Applying and developing these methods has become one of the most crucial and error-prone tasks in further analysis of brightfield images. In optical microscopy, especially when transmitted light and fluorescence microscopy are related to the specific cell structure segmentation, it is not hyperbolic to say that only a few distinct approaches about separating touching and overlapping cell structures represent a desirable performance with high efficiency and robustness.

Due to the fact that among all the rapidly developing pathological diagnosis methods, particular cell therapy has become one of the most popular research tasks. HeLa cell, which is the first continuous immortal cancer cell line, is commonly used in some processes of testing the toxicity and radioactive effects in human cells[1]. They do not suffer from programmed death since they maintain a version of the enzyme telomerase that represent regular decreasing length of the telomeres of chromosomes in the cell cycle. In addition, HeLa cells have been isolated and supported advances in most fields of medical research in the years.

We choose HeLa cells in specific cell tracking dataset [2] as our dataset, in order to detect HeLa cells in brightfield images and describe an approach to do cell detection and further analysis. Given a set of brightfield HeLa cell images in the cell cycle, we separate them into border, center and blank sessions as the labels. Patches are extracted from images after binarization. When they are distinguished and labeled, we utilize different filters as preprocess labels and carry on data augmentation in order to obtain abundant patches as our training dataset. We find that SVM is a desirable model for classification since it performs well in most datasets, and LeNet, which is able to respond to a part of the surrounding units, can also be applied in our experiment. Therefore, we prefer SVM and LeNet as our models to do classification and prediction.

Index Terms

HeLa cells, Cell tracking, neural network, machine learning, SVM, LeNet

I. INTRODUCTION

THE analysis of digital histopathological cell images in pathology diagnosis, drug discovery and cell therapy requires certain recognition technologies and main of them are manual. However, since there is a significant increasing demand for abundant accumulation of image, the accuracy and efficiency of manual methods are always debatable. We introduce some automated algorithms which have been developed rapidly and compare the rate of them with the traditional machine learning methods, addressing some potential problems present in cell segmentation, and possible solutions.

Cells are the structural base biological unit of all living organisms. In order to help to diagnose and assess the natural course of specific cells, applying certain abilities of detecting, segmentation and tracing cell activities involving cell motility, cell death, cell dividing cycle, determination of cell phenotype and other aspects are essential for further analyzing processes.

Several methods are utilized in high accuracy and temporal situations are frequently considered. In the past, 4'-6- diamidino-2-phenylindole (DAPI) was commonly used to quantitate cellular DNA in the center of the cell to track the whole cell in fluorescence microspectrophotometry[3]. However, since the DAPI dye is cell impermeant, the stain with higher concentration is able to enter living cells. In some occasions, when there is a set of cells exactly in the mitosis stage, the DAPI dye combines with the DNA quickly in interphase. As a result, the DAPI dye interferes with the normal cell dividing cycle of HeLa cells, and mutation probably occurs. That is the reason why DAPI and other fluorescent markers are always harmful to living cells and cannot be used in specific occasions of living cell segmentation.

In a way that does not harm living cells and interfere with cell cycle, defining and computing the rules as well as performing the training steps is actually challenging. There are also some limitations due to its specific nature: cell shape variability, touching and overlapping between cells, unsharp cell boundaries, etc. With extensive pre-processing steps, model-free approaches such as median filtering, thresholding, watershed segmentation and others are widely applied in specific cases of cell image analysis, though they sometimes fail to track the cell-cell contacts between multiple cells and do not provide a clear cellular shape.

We introduce a set of images of a group of HeLa cells[4] carrying cell dividing cycle, which are a kind of cancer cells commonly used in testing the radioactive and poisonous effects on human cells which are injected by some chemicals. A series of processes are involved in the HeLa cell mitosis cycle, which consist of prophase, metaphase, anaphase and telophase, and

two nuclei each containing the same number of chromosomes are present as a result[5]. Due to the fact that HeLa cells are cancer cells which have a short interphase in mitosis stage, a large proportion of the content in each cell is occupied by the nucleus. On the occasion that one cell starts to divide, it tends to become a circular shape and clearly stands out on the glass. A larger percentage of bright pixels are present in mitotic HeLa cells than non-mitotic HeLa cells.

In the method section, we present a complete system which is established to do cell detection, segmentation and carry on further analysis. First, border, center and blank patches are separated from the raw images and they are labeled with different colors. In the image preprocessing stage, several filters in Fiji are applied to the patches. After manually selecting, testing and comparing some patches, the most desirable size of the patches is determined. In order to increase the number of patches for training and testing, specific transformation is applied in the data augmentation phase. After that, two methods, which are support vector machines(SVM) and LeNet are selected and utilized as our model to do further training and testing. Finally in the visualization session, we illustrate a predicted image by the model, adjust the pixel value, classify the patched to different colors and draw a conclusion.

II. METHODS

A. Problem Statement

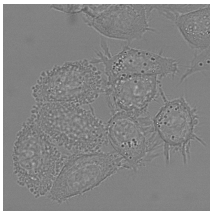


Fig. 1: A sample raw image

The task we mainly need to cooperate with in this project is cell detection and segmentation. Figure 1 indicates a cell which has more or less a circular shape, with a border and several nuclei. The shape of borders varies a lot: Some borders present are very smooth, whereas others have plenty bumpy places. Since it is difficult to observe the exact region for condensed DNA, we assume that the black dots in the HeLa cells act as the labeled nucleus. As a result, one cell might have only one or even none nucleus, while others may have five to six black dots as nucleus.Clearly, the significant variance of cell shape, border shape and number of nucleus of one cell is always the main reason why it is difficult to detect the location of the cells in one image properly. More precisely, it is always difficult to determine the exact location of the cell automatically when cell membrane and nuclei act as the strongest evidence for locating We propose a relatively easier solution to reduce the large uncertainty (see next section)

B. Idea

Before we go deep into our solution, a convention is required to be set: A cell (in an image) can be mainly divided to two components: 1) cell border 2) nucleus. As we mentioned before, the cells in an image could vary a lot: various shapes of cells and borders, and also different number of nucleus.

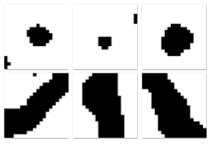


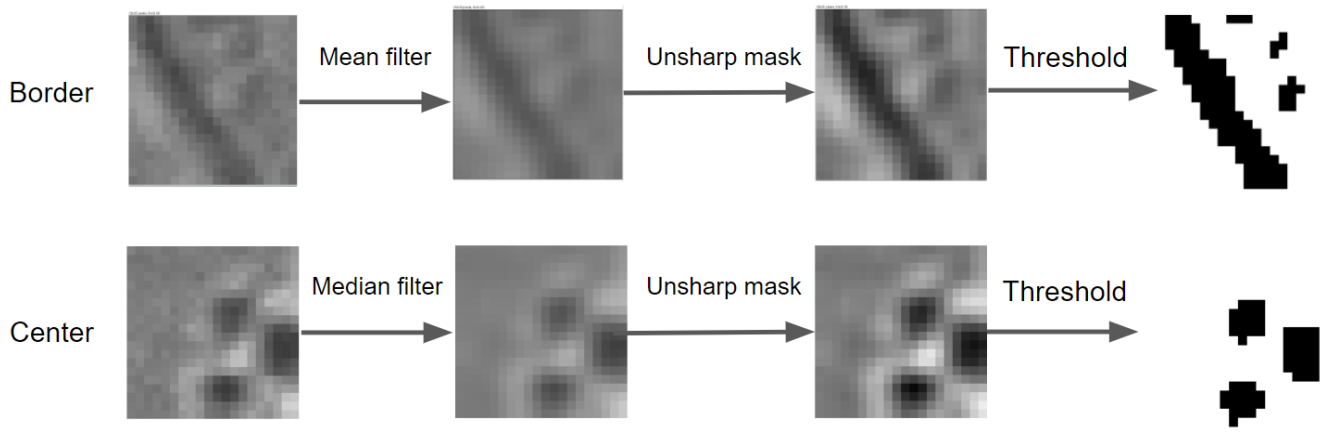
Fig. 2: nucleus and cell borders after binarization

However, if we decompose the cells like in figure 2 and compare their subparts, we will see that these subparts are looking similar. Basically, the nucleus in images are small solid circles whereas the the parts of cell border are like very thick solid lines. These subparts are relatively easier to detect or classify compared with a whole cell.

Inspired by many examples of image classification that have been very successful in the past such as MNIST handwritten digits classification [6] and also the relatively distinguishing features of the cell subparts, we come to an idea that we only detect or classify the patches of images as either **blank area** or **cell borders subparts** or **nucleus**. After that, we can mark these patches with different colors according to their classified classes.

C. Images Preprocessing

In Images Preprocessing part, we use Fiji to preprocess our raw images.In order to obtain great results to improve accuracy, we design and test many kinds of different flow. Then for the sake of high contrast, we use threshold, which filter the information in order to distinguish foreground and background well and also reduced the computing time and save storage space greatly. There are two preprocessing flow we choose to use.



D. Data Preparation

After image preprocessing, we need to create our own datasets. The first thing we did is to manually select about 20 representative patches for **blank**, **cell borders subparts** and **nucleus** respectively.

Notice that there is a trade-off between the patch sizes: The relatively big patches may contain cell borders and nucleus at the same time whereas the relatively small patches may not tell the difference between cell border subparts and nucleus. After practical experiments, we found that patches of 20x20 pixels relatively good for this classification.

Fig. 3: representative patches

1) Blank: The important feature of blank patches is "blank", i.e. the blank area occupies the most of the patch and the black area is little and only at the edge of the patch. One sample blank patch is given in figure 3 (the first patch).

2) Cell borders subparts: The cell borders (subparts) are also distinctive: Each border patch has a thick black solid line that penetrates that patch either horizontally or vertically, and the black area occupies the most of the patch. One sample border patch is given in figure 3 (the second patch).

3) Nucleus: The nucleus patches are even more intuitive: just several small black solid circles or ellipses at the center of the patches. One sample blank patch is given in figure 3 (the third patch).

E. Data Augmentation

After we manually select the patches above, we need to do some data augmentation on them to increase the number of patches.

We have done the following transformations: 1) horizontal and vertical shift 2) horizontal and vertical flip 3) random rotation 4) random zoom.

This step is not difficult, but one important issue we need to pay attention to is that after the transformation, the class of the patches should stay unchanged.For example: when we do horizontal or vertical shift on blank patches where the little black area is at the edge, the result may look like a border patch if we shift too much, since the new extra part due to the shift will directly copy the original edge pixels and thus look like a thick black solid line.

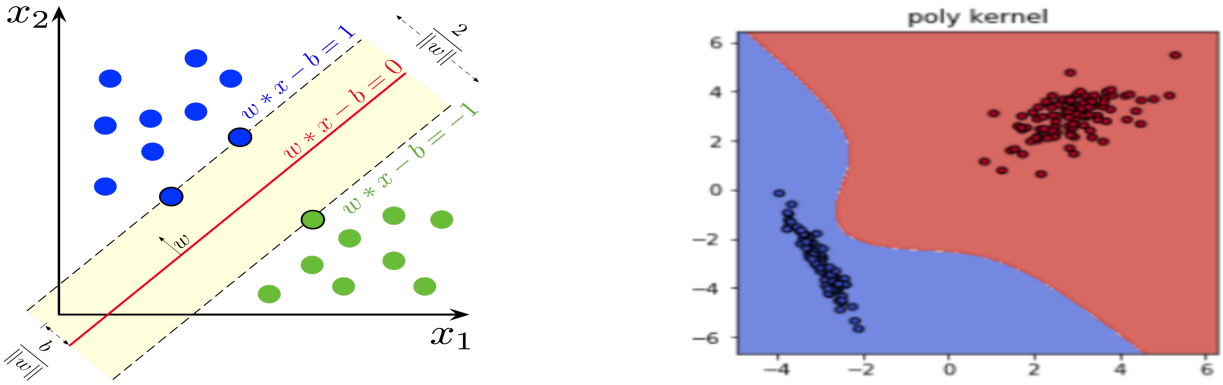
In practice we generate 45 new patches for one patch. And totally we have 714 blank patches, 1658 border patches and 1208 nucleus patches.

F. Models

After our dataset is created, we used them for model training. Here we use two well-known models: Support Vector Machine (SVM) and LeNet.

1) Support Vector Machine: [7]In machine learning, SVMs are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis[8]. In SVM, Kernel tricks are powerful transformation techniques that project data from its original space to a transformed space, hoping to make data more linear separable.After trying four different kernals, we eventually choose the polynomial kernel with the highest accuracy as our kernal. Preparing for the prediction,we trained our datasets and perfectly classified three labels,also obtained label classification by using SVM.

[8]The aim of the support vector machine algorithm is to determine a hyperplane in an N-dimensional space(N — the number of features) which classifies the data points clearly. In order to separate the two classes of data points, plenty of possible hyperplanes are selected to be chosen and tested. A plane with the maximum margin need to be found and confirmed, i.e the maximum distance between data points of both classes. Actually, the future data points are able to be classified in a



(a) SVM Model (b) SVM: Kernal = poly

Fig. 4: SVM Model AND POLY PLOT [7]

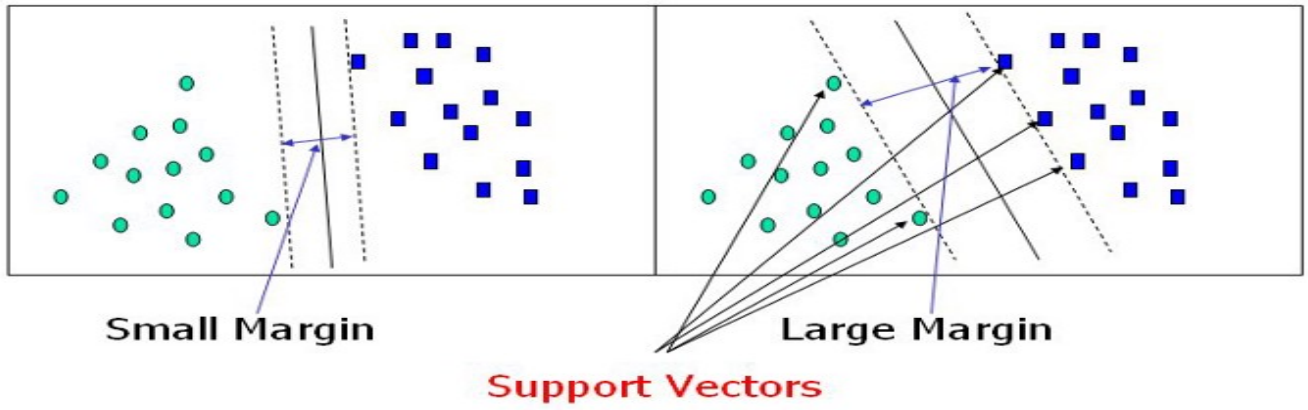


Fig. 5: Support vectors [8]

more reliable way when the margin distance is maximized. Due to the fact that it enhance the reinforce of the data points maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane.

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & if y * f(x) \geq 1 \\ 1 - y * f(x), & else \end{cases} \qquad c(x, y, f(x)) = (1 - y * f(x))_+ \tag{1}$$

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+ \tag{2}$$

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\begin{gathered} \frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k \\ \frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & if y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & else \end{cases} \end{gathered} \tag{3}$$

When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w) \tag{4}$$

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w) \tag{5}$$

2) LeNet: LeNet is a convolutional neural network structure proposed by Yann LeCun et al. in 1989. In general, LeNet refers to LeNet-5 and is a simple convolutional neural network [10]. In figure 6 you can see the classical architecture of LeNet.

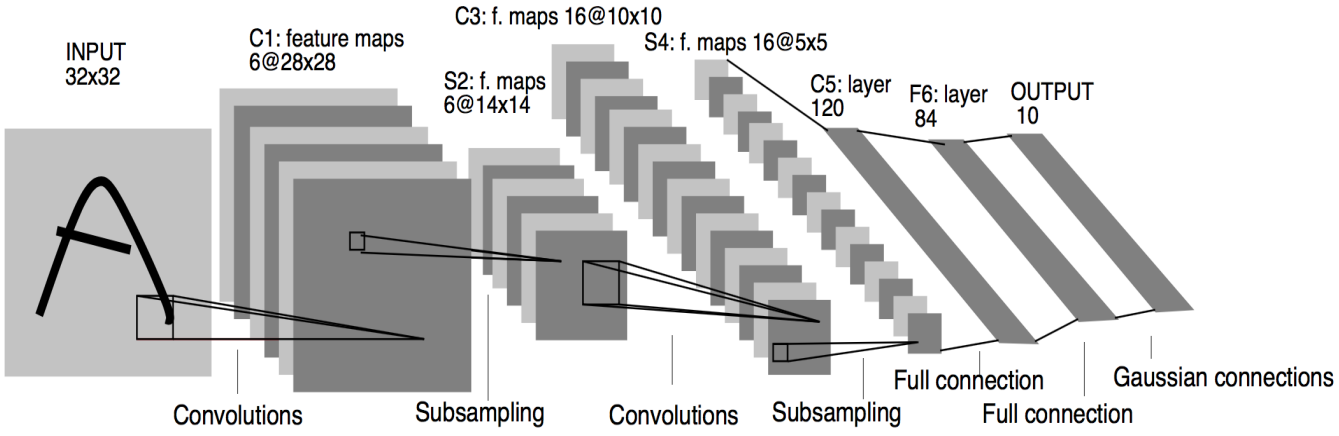


Fig. 6: LeNet Architecture

As shown in the figure: LeNet consists of seven layers: one input layer, three convolutional layers, two subsampling layers and one fully connected layer. Among them, C1, C3 and C5 are convolutional layers with the same kernel size 5×5 and stride 1. S2 and S4 are the subsampling layers with the same kernel size 2×2 and stride 2. The subsampling layers have generally two jobs, one is subsampling using max pooling, and the other is the activation. The activation function used in LeNet is sigmoid [11]. Since empirically some activation functions are better than sigmoid, such as ReLu function, we have used ReLu as the activation function in our project.

The most classic application of LeNet is simple digit images recognition, and that's also the reason why we choose LeNet because the two tasks are similar.

G. Visualization

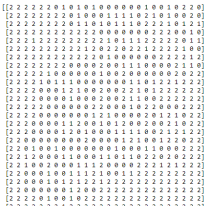


Fig. 7: predicted labels as 2D array

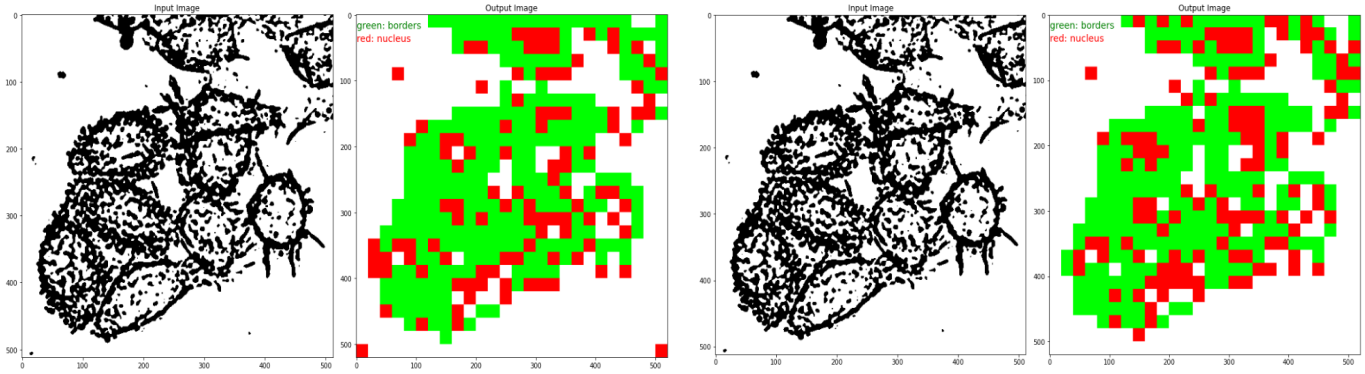
After the model is trained, we can use it to predict the new image patches and visualize the result. The general steps are: 1) Firstly we need to crop the input images into small patches. 2) Then we predict the class of these patches using our model. 3) Finally we mark these patches with different colors w.r.t their classes.

The first two steps are intuitive. As for the 3rd. step, we first try to display the predicted labels of the input image (patches) as a 2D array with width and height equal to the number of patches in each row and column respectively. That's to say, each entry in the matrix represents a patch in the original image, and the value of the entry indicates the class that patch is classified to. (0 for cell border subparts, 1 for nucleus and 2 for blank).

After that we can create a new image of the same size as the input image and set the pixel values of patches according to the value of the corresponding entry in the matrix. For us we mark the blank patch with white, the nucleus with red and the cell borders with green (see next section).

III. EXPERIMENTAL RESULTS

Here are the two images as our main results.1) SVM Visualization Result 2) LeNet Visualization Result



(a) SVM Visualization Result (b) LeNet Visualization Result

Fig. 8: Visualization Result

As we can see, for label 0(border) patches that are crowded together, SVM predicted better than LeNet, we think it's because our high accuracy SVM classification predict every kind of patches well.

In Fig 8 Cell 1, we predict this cell well, because this cell have unbroken boundary and black dots inside. However, in Fig 8 Cell 2, there are many irregular lines in the center of cells and many black dots around boundaries, they will definitely affect labeling and prediction. Many factors could impact our prediction, not only parameters we chosen and workflow we designed in the experiment but also tricky HELA cells' whose nucleuses are very difficult to label, so we choose whole black dots in the cells as our predict targets, which also will negatively impact our prediction.

In order to improve prediction results, here are some ideas to improve our experiment. 1) More suitable label patches size. Like pixels, a good deal of pixels means sharper image and higher accuracy, so if adjust patches size smaller, it would greatly improve accuracy. 2) Better batch size. [13] Appropriate parameters will improve accuracy, like batch size. Larger batch size will make the direction of descent more accurate and the shock would be more smaller. But in some cases, small batch size can get better results. 3) Threshold is a rapid way to sperate foreground and background ,but many information will be lost. There are many great preprocessing methods can separate foreground and background in low loss.It also can improve labeling and prediction accuracy.

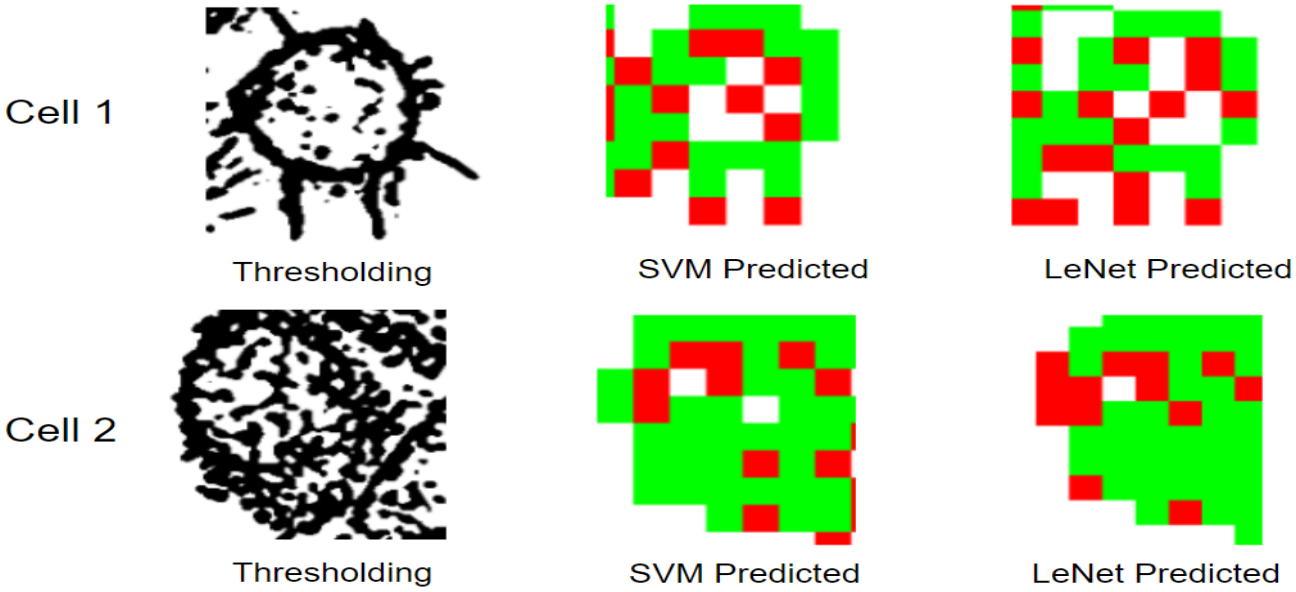


Fig. 9: Cells predicted detail

IV. CONCLUSION AND DISCUSSION

According to the presented experiment results, it is not hyperbolic to say that we draw a desirable conclusion and it is proved to be biologically meaningful. The method we used and developed can always be utilized in several segmentation of touching or overlapping sections in living cells in order to do further analysis in several pathological cell study fields. Based on our work, we consider that it can be used in further living cell tracking and detection of other organisms. Here are some of the future works based on the experiment results, hypothesis and related considerations.

1) Detection: Choosing a label from predicting matrix, and observe the other labels surrounding them in order to judge the label was correctly predicted or not. For example, if a patch is label 0(border) , the patches around the patch 0 all are label 2(blank), it probably is label 2(blank), but if the patches around the patch have 2-3 label 0(border) , it probably is label 0(border). After we get better predict matrixes, we would like to predict which part is cells. For example, if we find some label 1(center) gathering together ,we can detect other patches around these label 1(center) , if there are more than 3-4 label 0(border) around label 1(center) patches, then this part is probably portion of a cell.

2) Segmentation: on the occasion that cell detection is finished, the distance image is computed[14], while each foreground pixel stores the distance to the background pixel next to it. The replication stage, which stands for the location of each repeatedly entered foreground pixel, is established according to the number of repeats. However, when constructing the fitted GMs for specific locations of foreground pixels as the next segmentation, it is actually able to fit the pixel replication list. For these approaches with watershed algorithms, PR always gives a significant enhancement on them. GMs is also applied to fit the specific threshold image.

3) Cell Tracking: Based on our experimental results, [15]Attributed tracking graph(ATG)is illustrated since the pixel replicated elliptical shape model is applied in separating cell structures which touch together. The ATG represent extracted objects based on the features, time flow and the sequences of images.Without quantization, the following stages are repeated over the raw ATG images. And in the next phase, varying degrees of quantization is utilized: a) The normalized adaptive information distance (NAID) is used to calculate pairwise distances matrix. b) Gap spectral clustering is performed on this distance matrix, and after computing, the gap value is saved. In the next place, the clustering with the highest gap value, together with the corresponding feature subset is output as the summary.

REFERENCES

[1] John R.Masters(2002). Nature Review Cancer, HeLa cells 50 years on: the good, the bad and the ugly https://www.nature.com/articles/nrc775

[2] Dr. G. van Cappellen. Erasmus Medical Center, Rotterdam, The Netherlands, HeLa cells on a flat glass http://celltrackingchallenge.net/

[3] Annette W. Coleman, Mark J. Maguire, and John R. Coleman (1981) The Journal of Histochemistry and Cytochemistry Vol. 29, No. 8, pp. 959-968, Mithramycin- and 4 '-6-Diamidino-2-Phenylindole (DAPI)-DNA Staining for Fluorescence Micro Spectrophotometric Measurement of D NA in Nuclei, Plastids, and Virus Particles1 https://journals.sagepub.com/doi/pdf/10.1177/29.8.6168681

[4] What HeLa Cells Are and Why They Are Important https://www.thoughtco.com/hela-cells-4160415

[5] Joachim W, Lothar S, Marion C, Satoshi T, Thomas, C.(2003).Journal of Cell Biology, Chromosome order in HeLa cells changes during mitosis and early G1, but is stably maintained during subsequent interphase stages https://pubmed.ncbi.nlm.nih.gov/12604593/

[6] MNIST database from Wikipedia https://en.wikipedia.org/wiki/MNIST$_{database}$

[7] Support-vector machine from Wikipedia https://en.wikipedia.org/wiki/Support-vector$_{machine}$

[8] Support Vector Machine Explained-Theory, Implementation, and Visualization https://www.linkedin.com/pulse/support-vector-machine-explained-theory-visualization-zixuan-zhang

[9] Support Vector Machine — Introduction to Machine Learning Algorithms https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[10] LeNet machine from Wikipedia https://en.wikipedia.org/wiki/LeNet

[11] Sigmoid function from Wikipedia https://en.wikipedia.org/wiki/Sigmoid$_{function}$

[12] On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima https://openreview.net/pdf?id=H1oyRlYgg

[13] Separating Touching Cells using Pixel Replicated Elliptical Shape Models https://pubmed.ncbi.nlm.nih.gov/30296216/

[14] AUTOMATIC SUMMARIZATION OF CHANGES IN IMAGE SEQUENCES USING ALGORITHMIC INFORMATION THEORY https://ieeexplore.ieee.org/document/4541132