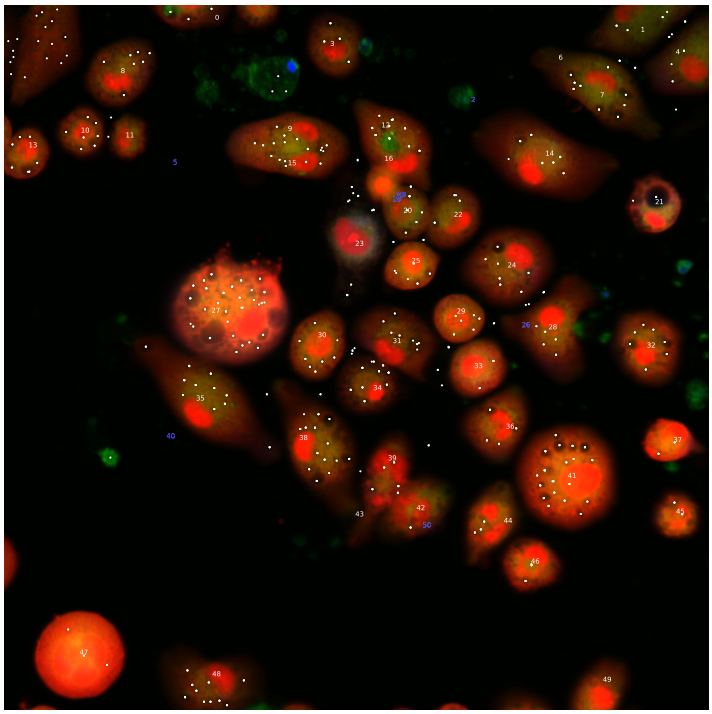
Automatic detection of cells and vacuoles through image segmentation.

Michael Smith

12th September 2017

Abstract

A watershed segmentation method, with cell nuclei acting as initial markers, is used to automatically detect cells from plates scanned by the IN Cell Analyzer 6000 [1, 2]. Morphological reconstruction by erosion is used to find dark spots in each found cell, and label them as vacuoles [3]. Finally, a shifted interquartile range is applied to the found cells' logarithmed sizes, to detect and remove outliers. A python script has been written to output an RGB image with all found cells and vacuoles labelled, and detected anomalies labelled in blue. An example is below. The script also outputs two log files per slide, describing attributes of each found cell and vacuole. This method takes around 5 seconds to scan a slide, and has been adapted to process slides in bulk from a given directory.



There is a git repository holding all the code used in this paper at `https://github.com/Smith42/cell-vacuole-finder`.

1 Methods

1.1 Data preprocessing

The slides that were made available are all generated with the use of an IN Cell Analyzer 6000 [2]. Each slide has had four different stainings applied to it. The nuclei are identified in the blue staining, cell membrane permeability in the green staining, mitochondrial toxicity in the orange staining, and cell plasma (along with vacuole locations) in the red staining. An example slide using the red, green, and blue stainings is shown in figure 1.

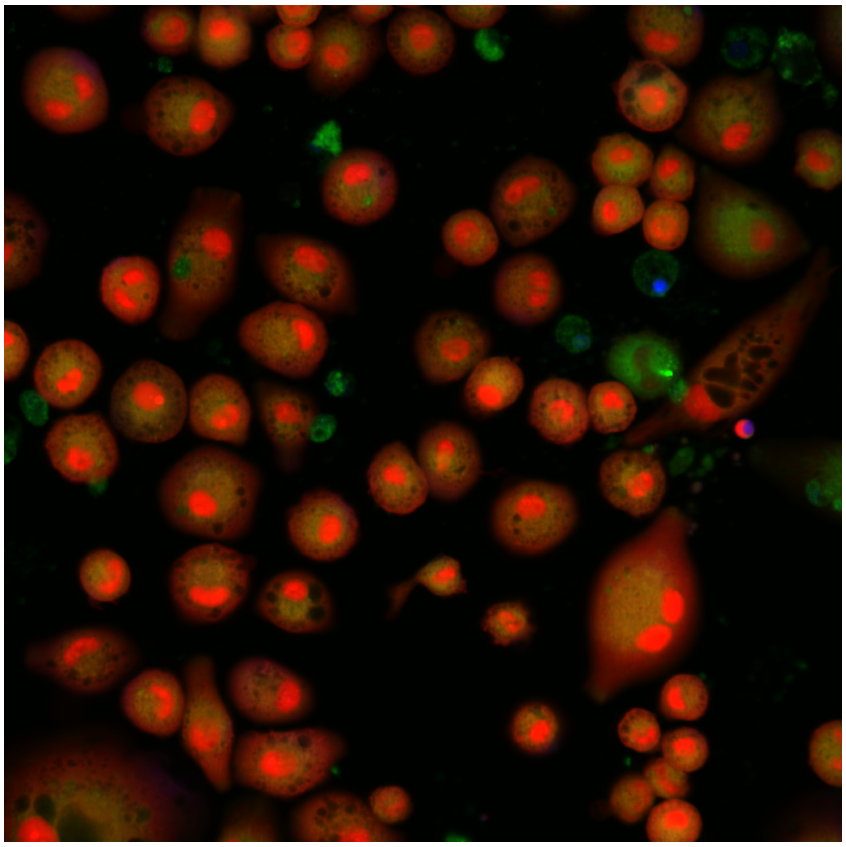


Figure 1: Example three channel image (red, green, and blue stainings) of a slide. Dark holes in the cell plasma are vacuoles.

1.2 Cell detection

Since the red staining contains information on the cell plasma, this channel has been used for cell detection through a watershed method [1]. The watershed method requires markers, to reduce oversegmentation.

The markers used for each cell are their respective nuclei, as found in the blue staining of each slide. To generate the markers, Li's minimum cross entropy

thresholding method is applied to the blue staining [4]. This separates foreground objects from the background into a binary image. The foreground cell nuclei are then labelled as separate objects, and set aside as markers.

The red staining is then rescaled so that every pixel with a value under the tenth, or over the 70th percentile is set at the value of the tenth, or 70th pixel. This reduces the contrast within each cell, and so reduces the possibility of large vacuoles interfering with the cell detection. As with the cell nuclei, the foreground cell objects are separated via a Li threshold applied to the rescaled red staining.

Once the mask for the cells, and the cell nuclei are found, a watershed segmentation is applied. Found cells are stored in a numpy array for vacuole detection.

1.3 Vacuole detection

Each cell in the cell array is fed through a vacuole detector function. This function uses morphological reconstruction by erosion to detect troughs in the image [3]. To generate a binary image, a mean threshold is then applied to the reconstructed cell [5]. Very small vacuoles are discarded, and the remaining found vacuoles are labelled.

1.4 Outlier removal

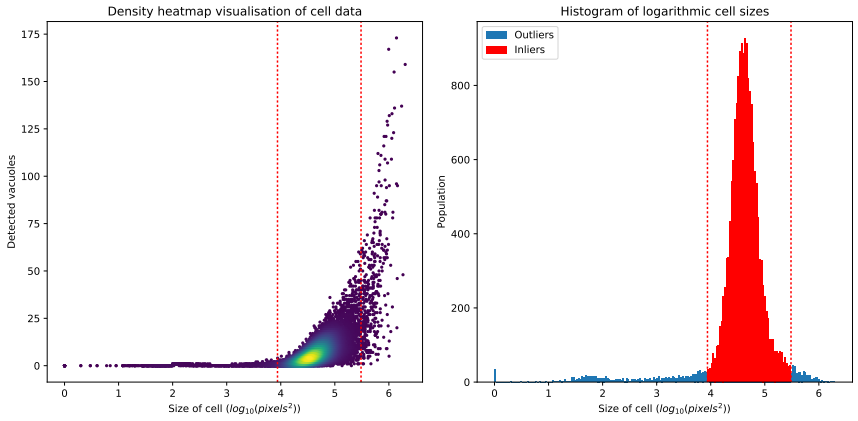


Figure 2: Interquartile range applied to logarithmed cell sizes. The upper and lower bounds are set as the 85th and 35th percentiles, since smaller cells were found to be more likely to be outliers. The left graph is a population density heatmap of each found cell in 630 slides. The right graph shows the population distribution of cell sizes.

Before outputting the found cells and vacuoles, outliers are detected and flagged. To achieve this, an interquartile range is applied to a large sample of cell sizes (figure 2). All cells found to be outside that range are flagged as outliers. Several different outlier detection techniques were tried[1], but were found to

[1] isolation forest, local outlier factor, and two-class t-SNE [6–8]

perform poorly. This is likely due to the use of unsupervised outlier classifiers on very high dimensional data. Performance could be improved greatly if labelled data were available. The labelled data could then be used as a training set for a supervised outlier classifier.

1.5 Output generation

The location of the found cell objects, and the relative location of each cells' found vacuoles are then used to overlay their positions onto an image of the original slide. In addition to this, the cells' sizes, positions, and number of vacuoles are outputted to a text file. The vacuoles' positions, corresponding cell, and sizes are also outputted to a text file.

2 Results

A selection of example output slides are shown below.

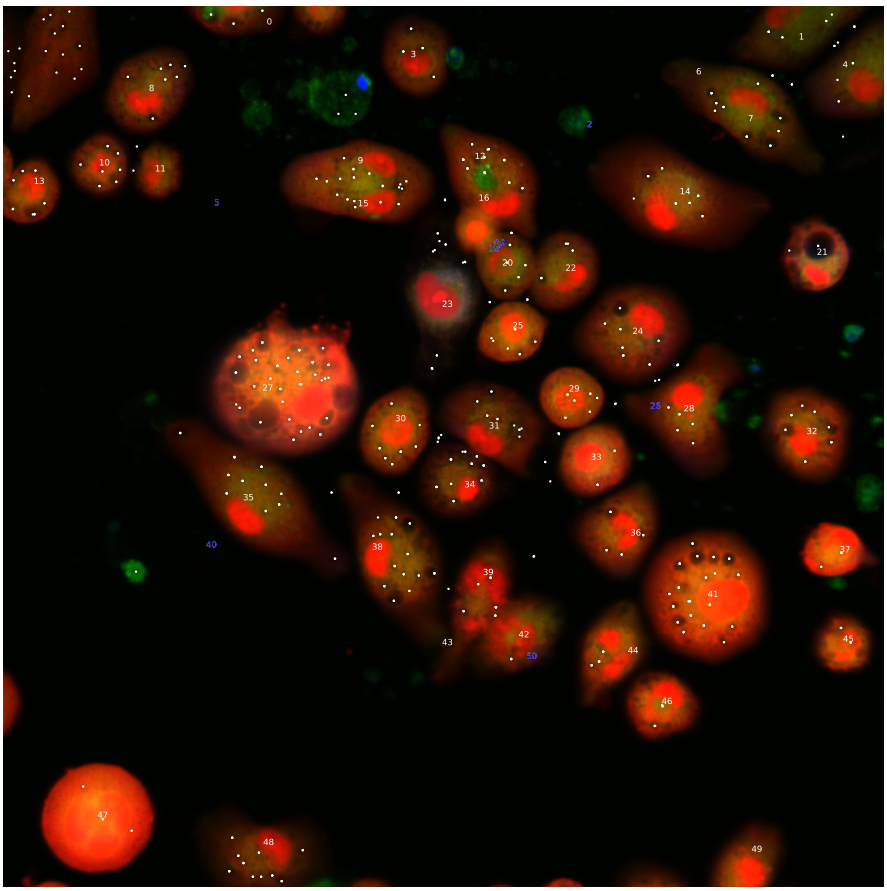


Figure 3: Example three channel output image (red, green, and blue stainings) of a slide. Found cells are labelled with a white number. Outlier cells are labelled with a blue number. Vacuoles are labelled with a white dot.

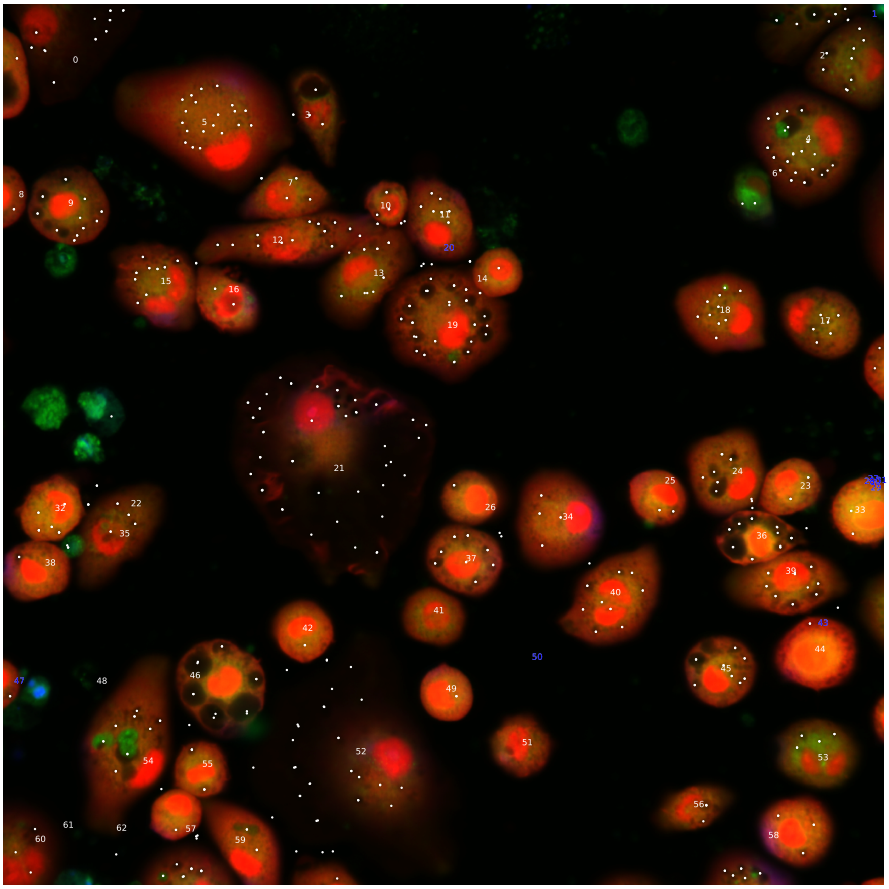


Figure 4: Example three channel output image (red, green, and blue stainings) of a slide. Found cells are labelled with a white number. Outlier cells are labelled with a blue number. Vacuoles are labelled with a white dot.

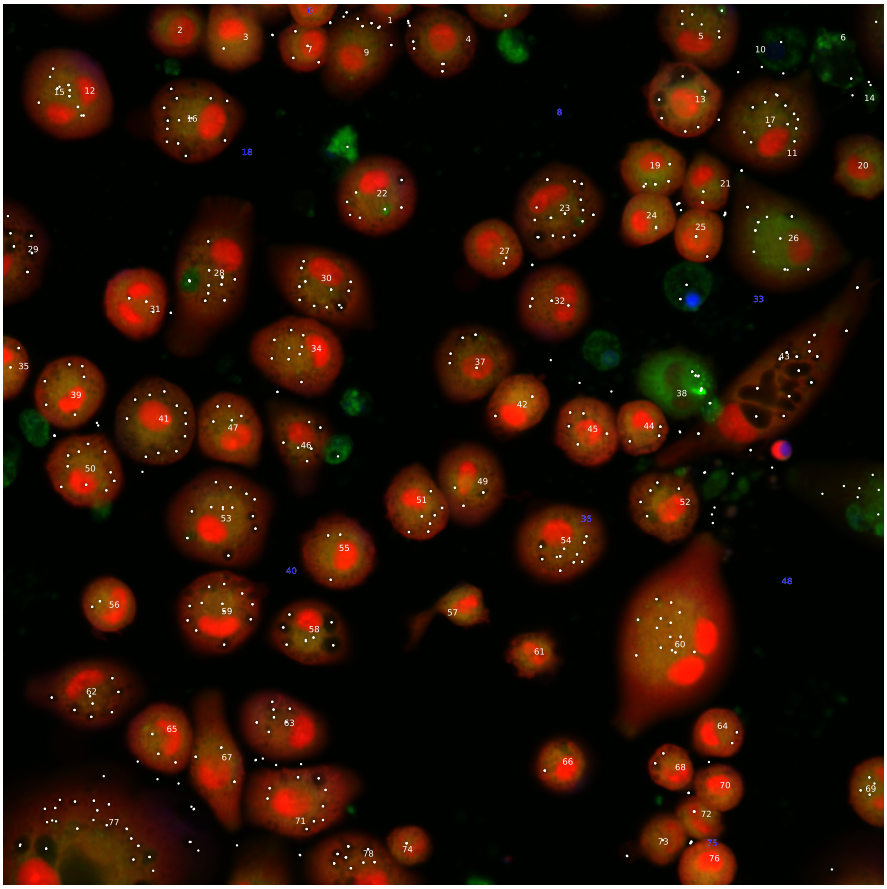


Figure 5: Example three channel output image (red, green, and blue stainings) of a slide. Found cells are labelled with a white number. Outlier cells are labelled with a blue number. Vacuoles are labelled with a white dot.

3 Discussion

Figures 3 through 5 show that the method outlined in this paper works relatively well. Unfortunately, it is difficult to compare the output of the algorithm to a ground truth, since there are no labelled slides available.

Using the nuclei as markers can introduce some anomalies. The detection of multi-nuclei cells as multiple cells, and the detection of noise in the blue staining as nuclei are two such issues with this marker choice. The noisiness could be partially mitigated by checking the nuclei found in the blue staining against peaks (likely nuclei) in the red staining. If labelled training data were available, a convolutional neural network based image classifier would be viable, and would likely increase the cell detection accuracy drastically.

Another approach to cell detection that could be tried is a general objectness algorithm, such as DeepBox, YOLO9000, or BING [9–11].

The vacuole detection algorithm sometimes detects vacuoles outside the boundary of the found cell. This could be remedied by masking the cell so that no pixels outside the cell boundary are fed to the vacuole detector. The brightness of the cells immediately surrounding the found vacuole could also be taken into consideration when deciding if the vacuole is anomalous or not, since vacuoles found outside the cell boundary will have darker surroundings. Again, a labelled vacuole dataset would be very useful, both for comparison to a ground truth, and for possible supervised learning methods.

4 Conclusion

A watershed segmentation method, with cell nuclei acting as initial markers, is used to automatically detect cells from plates scanned by the IN Cell Analyzer 6000 [1, 2]. Morphological reconstruction by erosion is used to find dark spots in each found cell, and label them as vacuoles [3]. Finally, a shifted interquartile range is applied to the found cells' logarithmed sizes, to detect and remove outliers. A python script has been written to output an RGB image with all found cells and vacuoles labelled, and detected anomalies labelled in blue. The script also outputs two log files per slide, describing attributes of each found cell and vacuole. This method takes around 5 seconds to scan a slide, and has been adapted to process slides in bulk from a given directory.

References

[1] Roerdink Jos B.T.M. and Meijster Arnold. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1, 2):187228, 2000.

[2] GE Healthcare Life Sciences. *IN Cell Analyzer 6000; Cell Analysis Redefined; User Manual*, aug 2011.

[3] L. Vincent. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, apr 1993.

[4] C.H. Li and C.K. Lee. Minimum cross entropy thresholding. *Pattern Recognition*, 26(4):617–625, apr 1993.

[5] C.A. Glasbey. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical Models and Image Processing*, 55(6):532–537, nov 1993.

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, dec 2008.

[7] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD*. ACM Press, 2000.

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[9] Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. DeepBox: Learning objectness with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015.

[10] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[11] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014.