Solving an object segmentation and recognition problem through Image Processing; Bacteria in Blood

1 Pre-Processing

The image is loaded and converted to grey scale. The image is resized to a height of 512 but retains the same aspect ratio. This is done by finding the division factor (dividing the rows by 512), then resizing to height of 512 with the width being columns divided by the division factor (**Appendix A**).

A produced histogram shows that the value range of pixels is at the lower end of the spectrum.

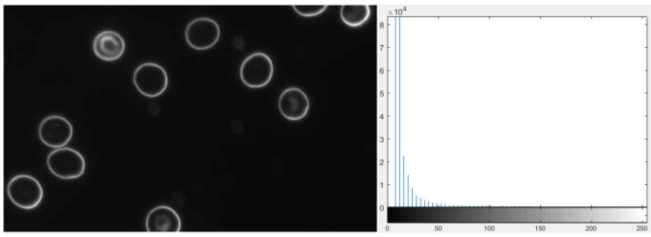


Figure 1 Resized greyscale image (not to scale) prior to enhancement

The image is enhanced by saturating the bottom 1% and top 1% of all pixel values, increasing contrast:

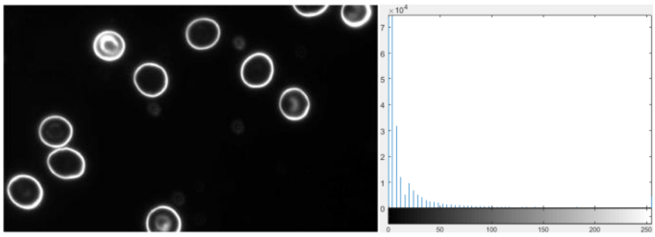


Figure 2 Increasing contrast through Matlab's imadjust() function (Chosen)

This is preferable to log transformation and histogram equalisation enhancement techniques, which instead of intensifying the contrast of the bacteria and blood cells against the background, highlights the noise in the background (**Appendix B**). This would be of more use if we were trying to find more "hidden" structures.

The image is binarized. The technique uses Otsu's method, which chooses the threshold value to minimize the intraclass variance of the black & white pixel threshold (Otsu 1979). The noisiest images appear to be image 03,04,07 and 08 (**Appendix C**).

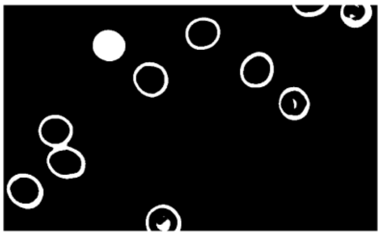


Figure 3 Binarized image

2 Edge Detection

The Canny (1986) edge detection method is used as it detects the circular objects efficiently. It localises well by minimising the distance between detected edge pixels and real edge pixels and has a relatively low error rate. (Ye 2021). The gradient is calculated using the derivative of a Gaussian filter. Two derivatives are computed which produces ridges. Outside of these ridges all other pixels are set to 0. Two thresholds are used to detect strong and weak edges. One point for each true edge is identified (Gonzalez, Woods 2018). This allows it to be more robust to noise and more likely to detect true weak edges (MathWorks 2022).

This can be compared to an older method such as Roberts (1963) which simply passes a pair of 2x2 convolutional masks over the image, where one mask is the other turned 90 degrees. As shown, it is much more sensitive to noise and does not produce edges as smooth as Canny.

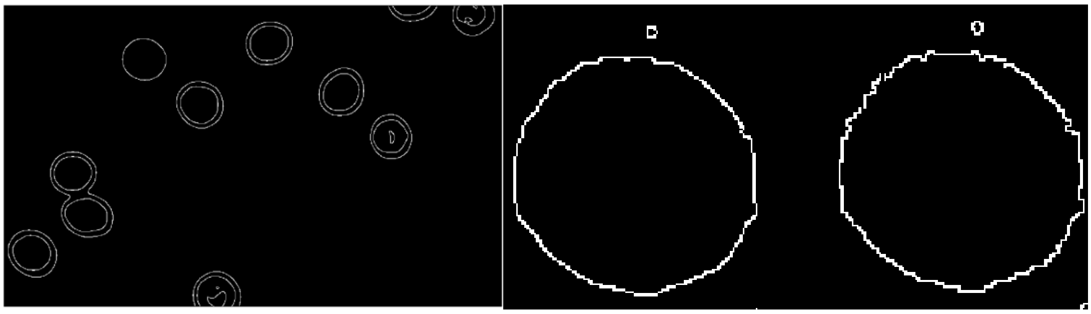


Figure 4 (Left to Right) "Canny" edge detection of cells; Canny (left) versus Roberts(right) edge detection

3 Simple Segmentation

A disk morphological structuring element of size 2 is created. Before being able to fill the cells, they need to be closed, which dilates the outside cells to help connect the sides then aims to bring them back to size through erosion. However, the cell boundaries at the perimeter are only seen as half-eclipses and so will not be able to close as easily as the rest.

In order to remedy the problem, the image is padded with white pixels, closed (with the disk element) and filled before removing the padding. The holes in the cells are filled with white pixels. According to MathWorks (2021) holes are “a set of background pixels that cannot be reached by filling in the background from the edge of the image”. The process occurs once for the perimeters of the X-axis (sides) and once for perimeters of the Y-axis (top & bottom) of the image. The padding must be removed intermittently, otherwise the entire image would be “closed” between the padded borders, rendering it white.

Small, connected components consisting of fewer than 300 pixels are removed such that only blood cells and bacteria remain, which are considerably larger (using MATLAB’s ‘bwareaopen’).

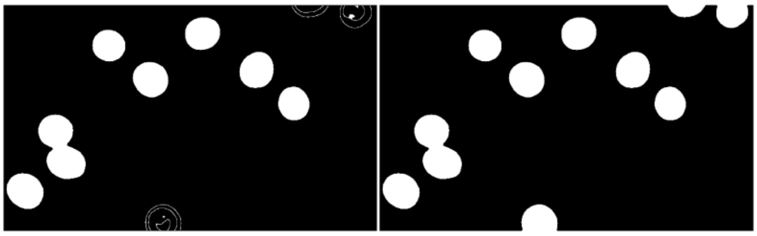


Figure 5 Segmentation without(left) and with(right) padding technique

4 Object Recognition

Region boundaries for remaining connected components are obtained and added to a list. The properties of each component are collected, with a focus on the area. For each boundary, if the area of the region is greater than 1000, the white pixels are filled red. If the area is less, the region is considered bacteria and so is shaded cyan.

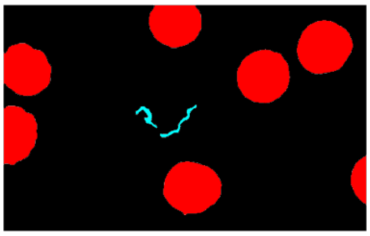


Figure 6 Object recognition in image 11

5 Robust Method

A more general robust solution is created to segment more accurately on all images in the dataset. Focusing mainly on image 06 of the dataset, I will discuss the alternative processing steps.

The image is loaded, converted to greyscale, and resized as per the previous solution. To smooth the noise in the image, a gaussian filter is applied with sigma 0.6. The difference is difficult to see with the naked eye, however for image 10 it is crucial to identify most blood cells.

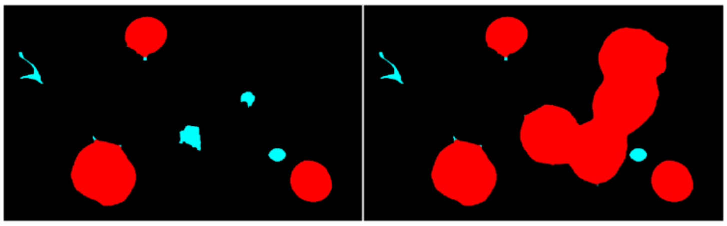


Figure 7 Image 10 end segmentation without gaussian filtering (left) and with filtering (right)

The contrast is enhanced yet again through saturating, but this time based on specified saturation limits 0.1 and 0.3, which happen to be ideal for the complete dataset.

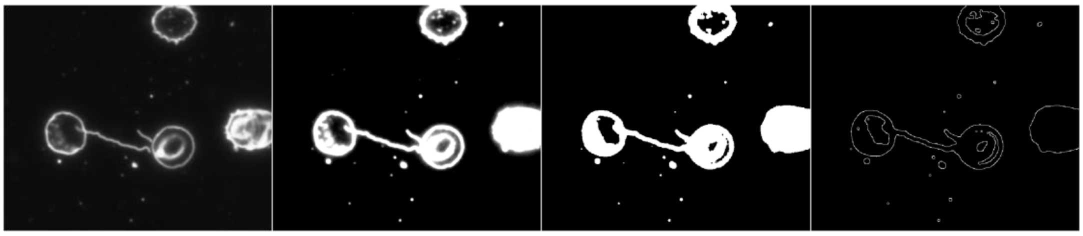


Figure 8 (Left to Right) Image 06 before; and after contrast adjustment; binarized; bounds located

The image is binarized this time using a global threshold of 0.4. The edges are found using the Canny Method.

The holes are filled using the padding/closed/fill/remove padding method of task 3. The major change of procedure occurs with segmentation. The image ‘F’ is eroded using a line of size 8 at 85 degrees as a morphological structuring element. A cell-only image is created ‘altF’ by opening with a disk element of 15 pixels instead of sole erosion. This harsh opening leaves the large circular structures.

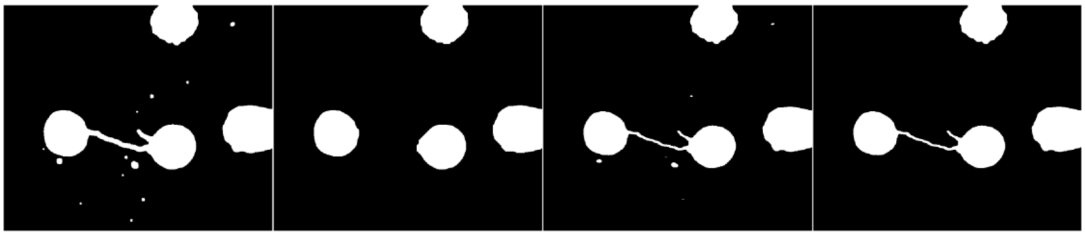


Figure 9 from left to right: filled image; 'altF' cell-only image; 'F' full image after slight erosion; 'F' with small components removed.

Small, connected components are removed of size 2000 or less for 'altF' and size 600 or less for 'F'. This is to avoid small bacteria from being removed in 'F'.

A new image is created based on the union of 'altF' and 'F'. On a pixel-by-pixel basis, if the pixel has a value in 'F' and 'altF', it must be a Cell, and is set to value '1'. If the pixel is only present in 'F', it must be bacteria, and is assigned value '2'. The background will remain with '0' pixel values. The image is coloured based on these logical pixel values and the segmentation image is created (and saved to an output file). This method works as a general solution to the problem. However, as forfeit, some cells that were identified in images through the first (1-4) method are now seen as bacteria. This is due to their view being obstructed at the perimeter and their vulnerability to erosion. Possible improvements are noted in **Appendix D**.

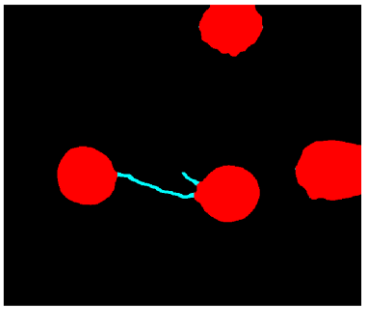


Figure 10 Image 06 final segmentation. Red = Blood Cells; Cyan = Bacteria

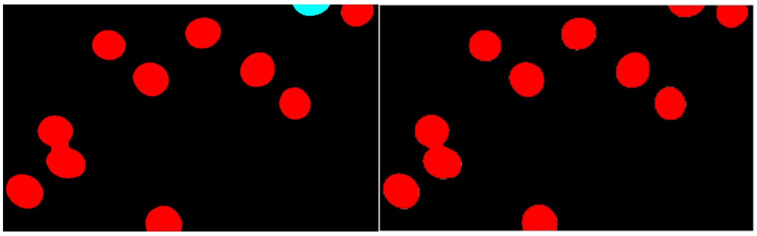


Figure 11 "Robust" solution works generally better at spotting cells and bacteria, but accuracy is forfeited for particular images (Image 01 Robust Method; Image 01 Original Method)

6 Performance Evaluation

The "Robust" Method is evaluated by comparing the ground truth images with the resulted segmented images, using Dice Score, Precision and Recall. Image differences are visualized in **Appendix E,** which supports the DICE score similarity. Precision measures the number of predicted pixels that have a matching ground truth annotation, while recall measures the number of captured positive predictions (Jordan 2018). The metrics returned are as follows, calculated for each class (an alternative would be averaging between the classes):

Table 1 Scoring metrics for "Robust" Segmentation Solution

IMAGE NO.	OBJECT TYPE	DICE	PRECISION	RECALL
1	Cell	0.95141	0.96159	0.94145
2	Cell	0.93893	0.98091	0.9004
3	Bacteria	0.72065	0.56636	0.99046
4	Cell	0.67832	0.70116	0.65692
	Bacteria	0.46741	0.31808	0.88103
5	Cell	0.73719	0.59724	0.9628
	Bacteria	0.70419	0.67054	0.7414
6	Cell	0.94517	0.92609	0.96505
	Bacteria	0.66272	0.88179	0.53084
7	Cell	0.79039	0.65848	0.98837
	Bacteria	0.55942	0.40248	0.917
8	Cell	0.6396	0.63294	0.6464
	Bacteria	0.12694	0.1406	0.1157
9	Cell	0.82203	0.70083	0.99392
	Bacteria	0.42822	0.38761	0.47834
10	Cell	0.90298	0.82754	0.99356
	Bacteria	0.25485	0.41931	0.18306
11	Cell	0.91312	0.84425	0.99423
	Bacteria	0.7694	0.73777	0.80386
12	Cell	0.5842	0.58933	0.57917
13	Cell	0.73424	0.58107	0.99707
	Bacteria	0.74144	0.59703	0.978
14	Cell	0.94513	0.95015	0.94017
15	Cell	0.82214	0.70817	0.97984
	Bacteria	0.67311	0.62778	0.7255

Images 07, 08 and 09 seem to have the worst scoring measures, indicating that the segmentation method is inefficient for these and would need to be revised. This is understandable as can be seen in **Appendix E**.

7 REFERENCES

Barla N (2021) The beginner's Guide to Semantic Segmentation, v7labs. Available from: The Beginner's Guide to Semantic Segmentation (v7labs.com). [Accessed 28 January 2022]

Canny J (1986) A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, 8(6), p679–698. Available from https://pubmed.ncbi.nlm.nih.gov/21869365/. [Accessed 30 January 2022].

Gonzalez RC, Woods RE (2018) Image Segmentation; The Canny Edge Detector. In: *Digital Image Processing*. Pearson, New York. P729-733. Available from: https://ebookcentral-proquest-com.proxy.library.lincoln.ac.uk/lib/ulinc/reader.action?docID=5573669#. [Accessed 30 January 2022].

Jordan J (2018) Evaluating image segmentation models. Available from: https://www.jeremyjordan.me/evaluating-image-segmentation-models/. [Accessed 30 January 2022].

MathWorks (2021) "imfill" documentation, *Mathworks*. Available from: https://uk.mathworks.com/help/images/ref/imfill.html#d123e139592 [Accessed 20 December 2021].

MathWorks (2022) "edge" documentation, *MathWorks*. Available from: https://uk.mathworks.com/help/images/ref/edge.html. [Accessed 07 January 2022].

Otsu, N (1979) "A Threshold Selection Method from Gray-Level Histograms.*" IEEE Transactions on Systems*, Man, and Cybernetics. Vol. 9, No. 1, p62–66.

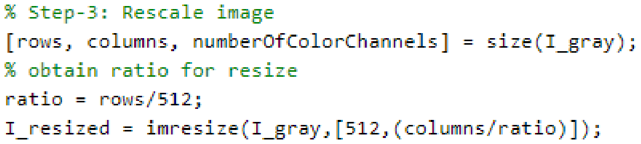
Roberts (1963) *Machine Perception of Three-Dimensional Solids*, Garland Publishing , New York. Available from: https://www.researchgate.net/publication/220695992_Machine_Perception_of_Three-Dimensional_Solids. [Accessed 07 January 2022].

Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. Lecture Notes in Computer Science, vol 9351.

Ye X (2021) Lecture 7, *Image Segmentation*. CMP3108M Image Processing, University of Lincoln. Delivered November 2021.

8 Appendix

A – Resizing Images whilst maintaining resolution



```
% Step-3: Rescale image
[rows, columns, numberOfColorChannels] = size(I_gray);
% obtain ratio for resize
ratio = rows/512;
I_resized = imresize(I_gray,[512,(columns/ratio)]);
```

B - Log Transformation & Histogram Equalisation

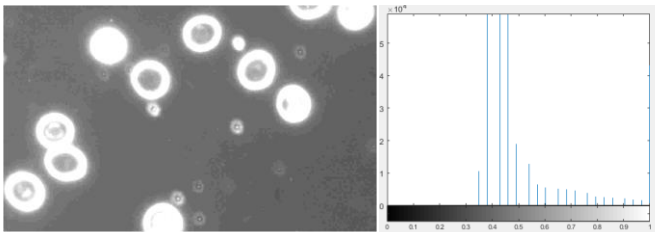


Figure 12 Image after log transformation (Not chosen)

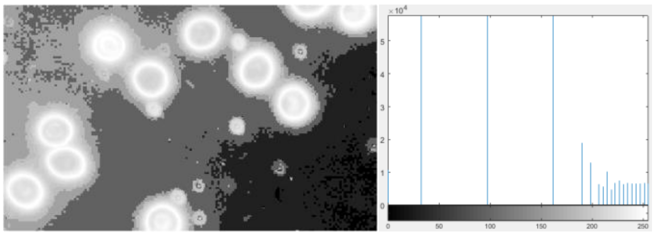


Figure 13 Image after histogram equalisation (Not chosen)

C – Noise in Binarized images

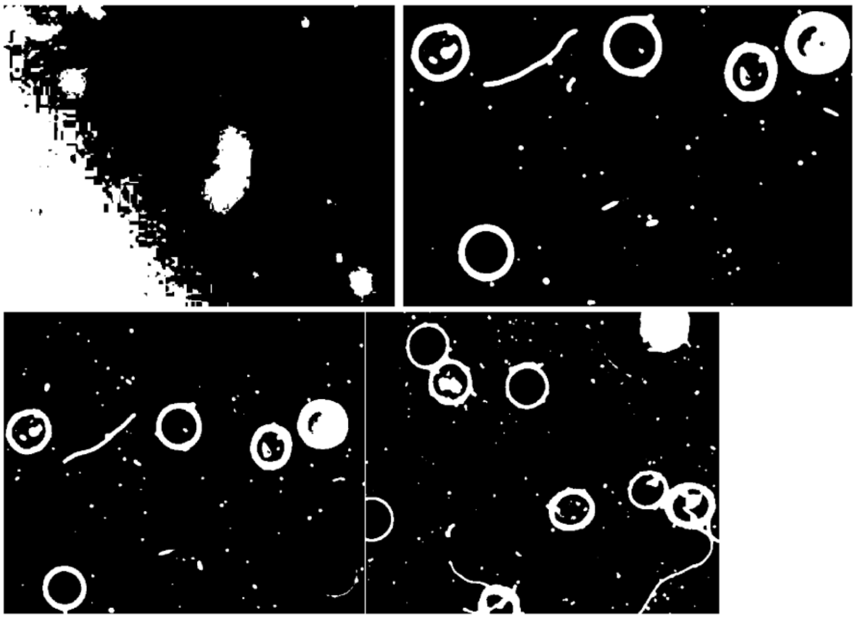


Figure 14 Noise in images 03, 04, 07, 08 after binarization by original method

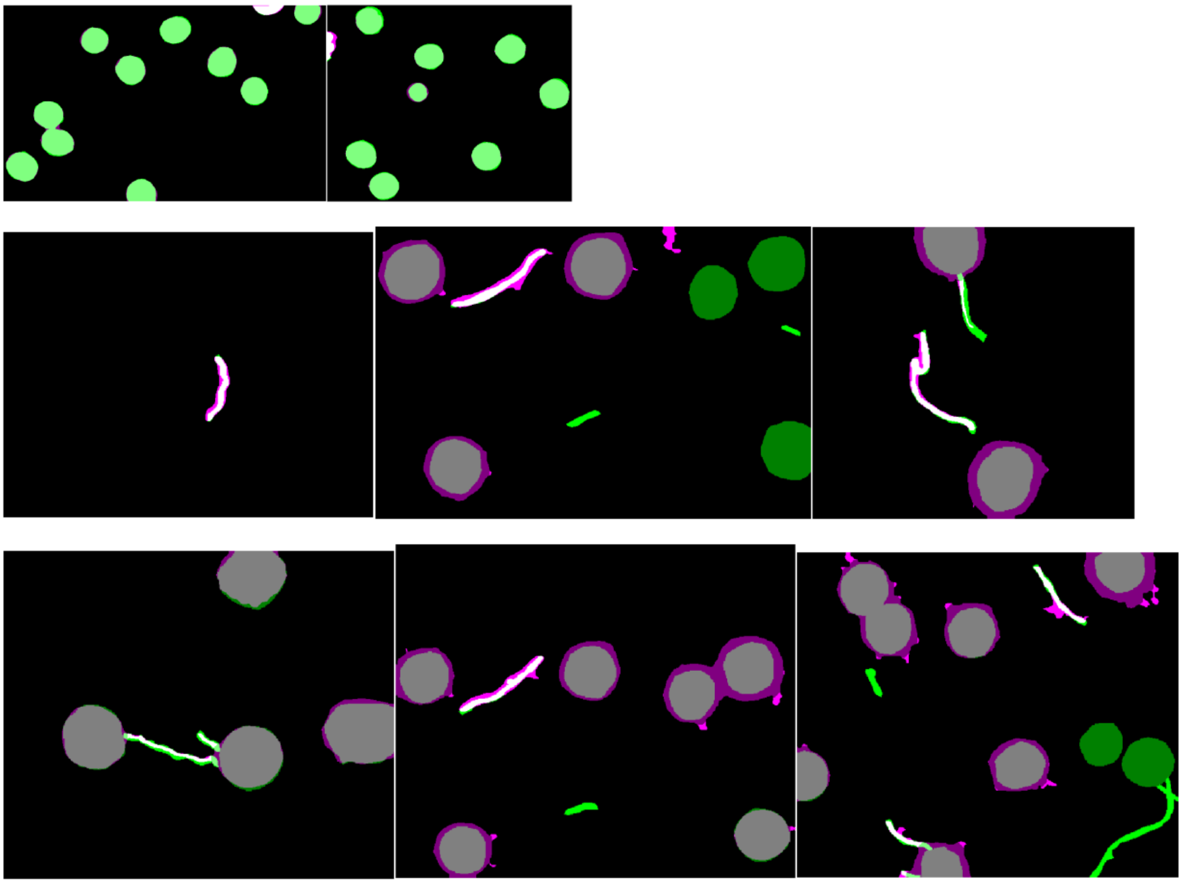
D – Possible improvements to robust method

The "robust" solution has no concept of component features unlike the original segmentation method. It relies on blood cells being large and round in nature which are invulnerable to opening within a threshold. A more efficient solution may include the feature extraction method incorporated into the harsh opening and union method. It could examine properties such as circularity, minor and major axis length of connected components. More initial filter methods could help avoid the disappearance of some cells, and a more dynamic contrast adjustment could identify cells hidden in the background.

Another way of reaching the most efficient solution could be through using machine learning (Computer Vision). Semantic segmentation would assign every pixel a value whether it is a blood cell or a bacterium. It begins with classifying an object, drawing a bounding box around it and then creating a segmentation mask by grouping its pixels.

Semantic Segmentation can be carried out by passing the image through a trained convolutional neural network. Due to CNNs reducing the size of the image, the image is "unsampled" using interpolation after it is sampled- using deconvolutional layers at the end as opposed to ending with a fully connected classification layer. This leads to an encoder-decoder structure (Barla, 2021). The U-Net is an example of an encoder-decoder CNN (Ronneberger, Fischer, Brox 2015) and could potentially be used for the semantic segmentation of blood cells and bacteria in blood.

E – Visual differences between ground truth and segmented images using "Robust" Method



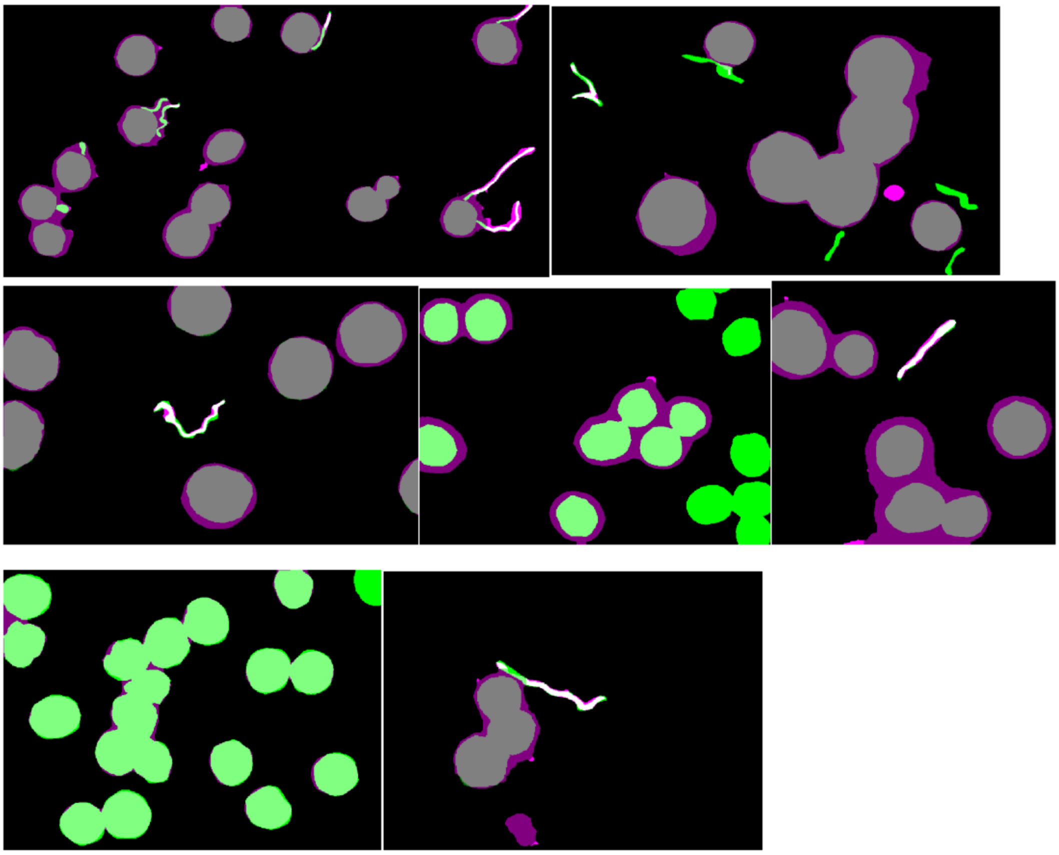


Figure 15 Visual differences between segmented images by "Robust" method and ground truth, in chronological image order.