Cell Analysis Website Standard Operating Procedure

Purpose

Understand the applications for the cell analysis application, the application's tools, and the underlying mechanisms used to transform raw images into quantitative data.

Overview

Why Build This?

The goal of this application is to provide an alternative to other scientific image analysis tools, like ImageJ. While used extensively by the scientific community, ImageJ fails to take advantage of modern algorithms. Additionally, ImageJ's user interface is outdated by modern standards and unintuitive to the average user.

Cell Analysis Capabilities:

This website contains various tools that are useful for obtaining data from cell imaging. There are three main outputs that result from the processes used by the website:

1. **Number of Cells.** This is determined by a border following algorithm created by OpenCV (cv2) called findContours(). After contours are found, the number of cells is determined by the number of contours. However, if cells are clumped together, they will only be counted as a single cell. To mitigate this, the website will recognize when a contour is larger than a specified size. If the contour is larger, the website will divide the contour area by the average size of the cell, giving a close estimation of the actual number of cells.

2. **Cell Area (by contours).** This data also is a product of findContours(). The area of all contours is added together and output.

3. **Cell Area (by threshold).** This also returns area, but it does this using a more traditional method. Instead of using contours, the area is determined by the value of each pixel. If the pixel is within the thresholds defined by the user, it will be counted towards the area. This is akin to the method used by ImageJ

Transformations:

Image Processing. This is where this application excels. Compared to ImageJ, there are substantially more options for pre-processing an image. Most significant of these transformations include kernel convolution methods like Sobel, Canny, and Laplace. These transformations are edge detection algorithms used across the industry for self-driving cars, motion tracking, spatial mapping, and various other computer vision algorithms. To learn more about these, I'd suggest checking out this lesson.

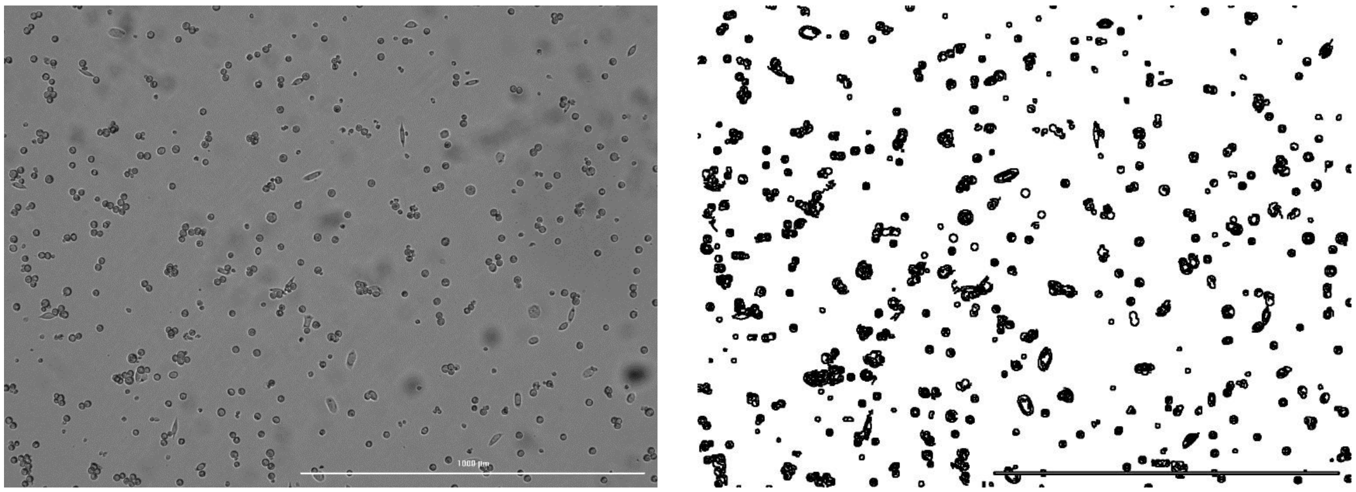


Figure I: Base image (left) compared to after a Canny convolution (right)

There are other processing methods than Sobel, Canny, and Laplace, but they have not been fully implemented yet. Block Normalization is an outdated method and will likely be removed later on.

Morphological Operations. There are four operations that fall under this category. Open, Close, Dilate, and Erode. More info can be found here, but this is an overview.

- Open: Removes noise in an image
- Close: Fills in small holes/gaps in objects in the image
- Dilate: Thickens the edges of objects in the image
- Erode: Thins the edges of objects in the image

By default, one iteration of open and close is applied.

User Interface

Pages:

There are two pages in the application: Cell Analysis and Analyzer Pipeline. Cell Analysis can be used to go through each picture one-by-one. However, doing this can be very tedious, especially if all the images look similar. The Analyzer Pipeline can be used to analyze an entire batch of images.

Page 1: Cell Analysis

There are three sections of the 'Parameters' container. The first section is the uploader. Click to browse or drag and drop the image of choice. The second section, "Cell Characteristics", is used to describe the cell's size.

Minimum Area to be Considered Cell. This defines how small of an area will be considered a cell. If set too small, small pieces of debris or noise could be counted as a cell.

Average Size of Single Cell. This defines the average size of the cell. When clumps of cells are detected, the area of that clump will be divided by this value to determine the amount of cells in the clump

Max Size of Cell. A threshold value for when a contour is considered larger than a single cell. After passing this threshold, the area of the contour will be divided by the average size of the cell.

Minimum Area, Average Size, and Max Size of the cell can all be viewed visually on the first image in the second column after generating. Use this to refine your inputs.

Intensity Thresholds. If the pixel is within the thresholds defined by the user, it will kept in the picture. If the pixel falls outside of the threshold, it will be turned black.

Scaling. Set this value to convert pixels to micrometers. Defaults to 0.595

Image Processing:
The last section of the "Parameters" expander is for image processing. This is where the user can select settings for the previously described transformations.

Processing Method. Selects between different image processes, as described previously

Apply Morphological Transformations. This will open up a menu for the four operations previously described: Open, Close, Dilate, and Erode

Use Fluorescence. This inverts the image. Brightfield cells are darker than the background, whereas fluorescent cells are lighter than the background. The fluorescence setting also changes the green overlay to blue.

Minimum Hole Size for Subtraction. When defining contours, holes within the contour can be subtracted (like subtracting the donut hole from the donut). If set too low, noise may be incorrectly calculated as a hole. When set too high, holes won't be subtracted. Holes are highlighted in red in the overlayed image.

Kernel Size. This is a setting used for the processing method (Sobel, Canny, and Laplace). A 'kernel' is just the size of the matrix used for convolution. Think of the kernel like a paintbrush. The larger the paintbrush, the lower the detail but the smaller the noise. A kernel size of 3 is generally the best and is set as default.

Morph Iterations. When the checkbox is checked for morphological transformations and a specific operation is chosen, a slider can adjust how many iterations of the operation are run.

Results:

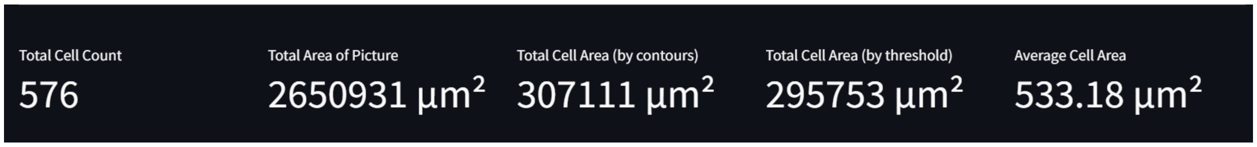
The outputs described in a previous section will be displayed.

- Total Cell Count
  - A total number of cells calculated by number of contours and specified cell sizing
- Total Area of Picture
  - All pixels in the image, converted to um. Helpful for determining % of coverage
- Total Cell Area (by contours)
  - Determined by area inside of all the contours added together
- Total Cell Area (by threshold)
  - Determined by total white pixels in the morphed/masked image
- Average Cell Area
  - Contour cell area divided by number of cells

Something to note: If the contours fail to capture cells effectively, look at the morphed image (or masked image if not using morphological operations). If the white in the image captures the cell area well, but the contours do not, you can still use 'Total Cell Area (by threshold)".

While the contours method is helpful, the threshold area is far more robust, especially in highly confluent plates.

A good way to measure how well the application works is by comparing the Total Cell Area (by contours) to the Total Cell Area (by threshold). These two areas should be relatively close to one another under normal conditions.



Total Cell Count	Total Area of Picture	Total Cell Area (by contours)	Total Cell Area (by threshold)	Average Cell Area
576	2650931 μm^2	307111 μm^2	295753 μm^2	533.18 μm^2

Figure II: Metrics data calculated from image

After the outputs, six images will be displayed:

- Original image
- Image with the three cell parameters (as previously described)
- Processed image (after Sobel, Canny, Laplace)
- Masked image (processed image + applied thresholds)
- Morphed image (masked image + morphological operations)

Overlayed image (original image, contours draw in green (or blue if fluorescence), holes drawn in red)

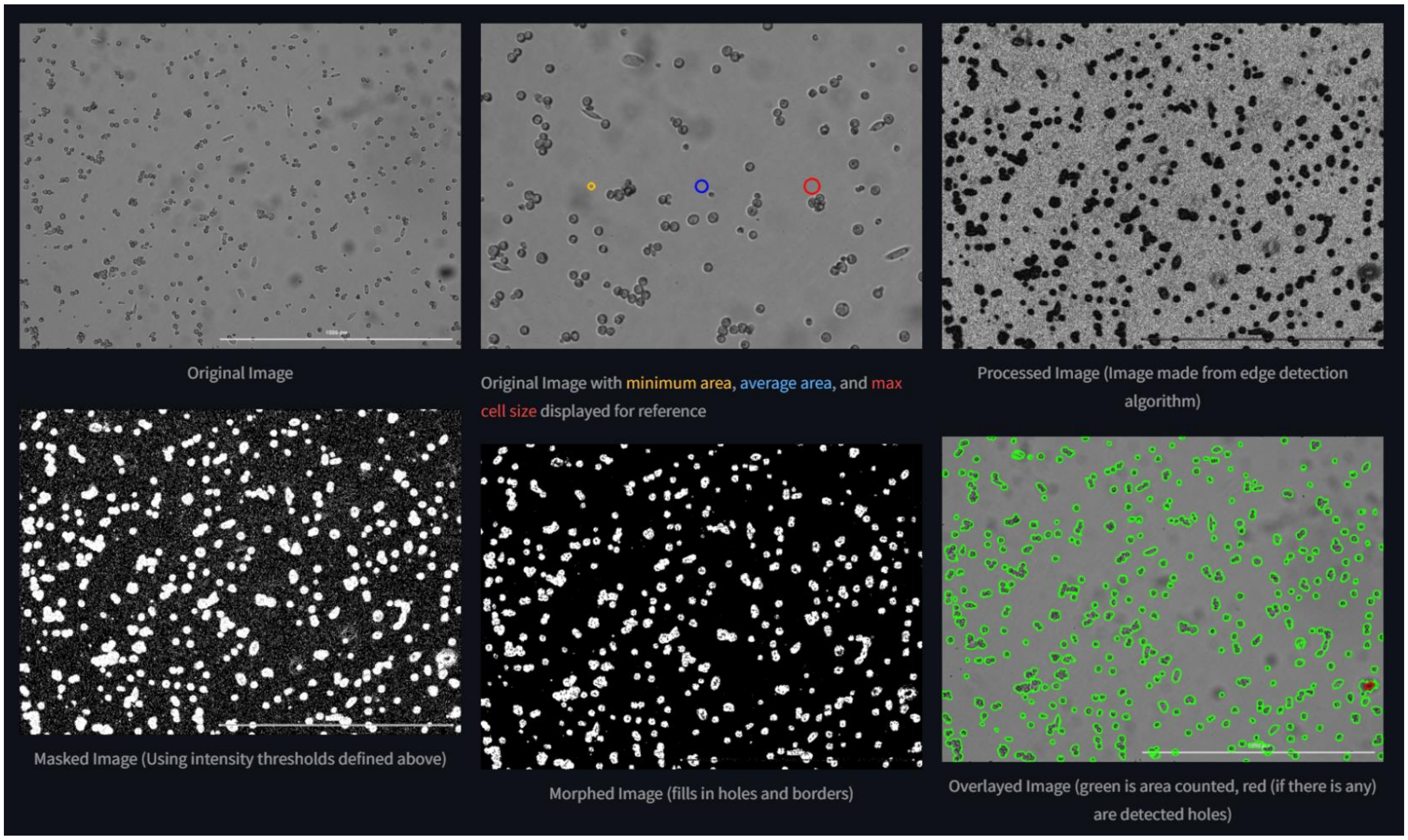

Original Image
Original Image with minimum area, average area, and max cell size displayed for reference
Processed Image (Image made from edge detection algorithm)
Masked Image (Using intensity thresholds defined above)
Morphed Image (fills in holes and borders)
Overlayed Image (green is area counted, red (if there is any) are detected holes)

Figure III: Displayed images from each step in the process

Lastly, there is a slider with the original image and overlayed image. This is helpful for viewing how well the contours surround the cells.

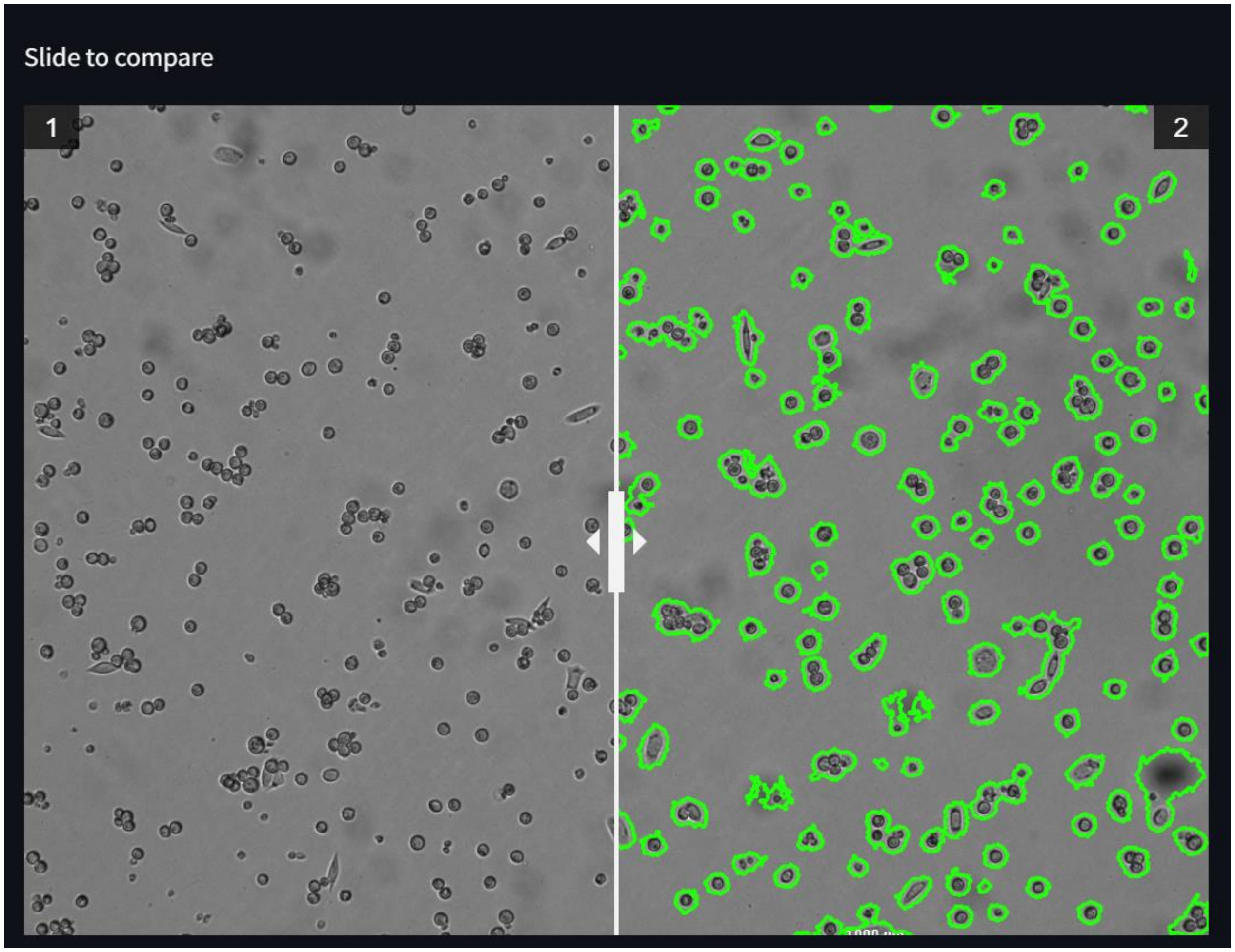


Figure IV: Slide-to-compare feature for viewing contour accuracy

Page 2 – Analyzer Pipeline:

This page is nearly identical to the first page, but contains a few more resources to speed up the process.

Settings Uploader. After running the pipeline, all of the files generated by the application can be downloaded as a zip file. One of the files contained in the zip file is settings.csv. This can be uploaded in the “Upload Settings” expander at the top of the page for a future batch. This way, there is no need to manually keep track of and change the parameters to match what was done previously. Uploading the settings file will automatically hide the “Parameters” expander.

Run Test vs. Run Batch. After uploading multiple images, the parameters set can be tested by clicking “Run Test”. The output will look identical to the first page. If the user is satisfied, they can then click “Run Batch” to begin the pipeline.

Run Batch:

After running the batch, two output images are displayed for each input image. The first image is overlayed contours on the original image. The second image is the morphed/masked image. If using contour data, the left is most important. However, like previously stated, thresholding is more robust. So, in the case that contouring doesn’t work well, the user can use the thresholding method.

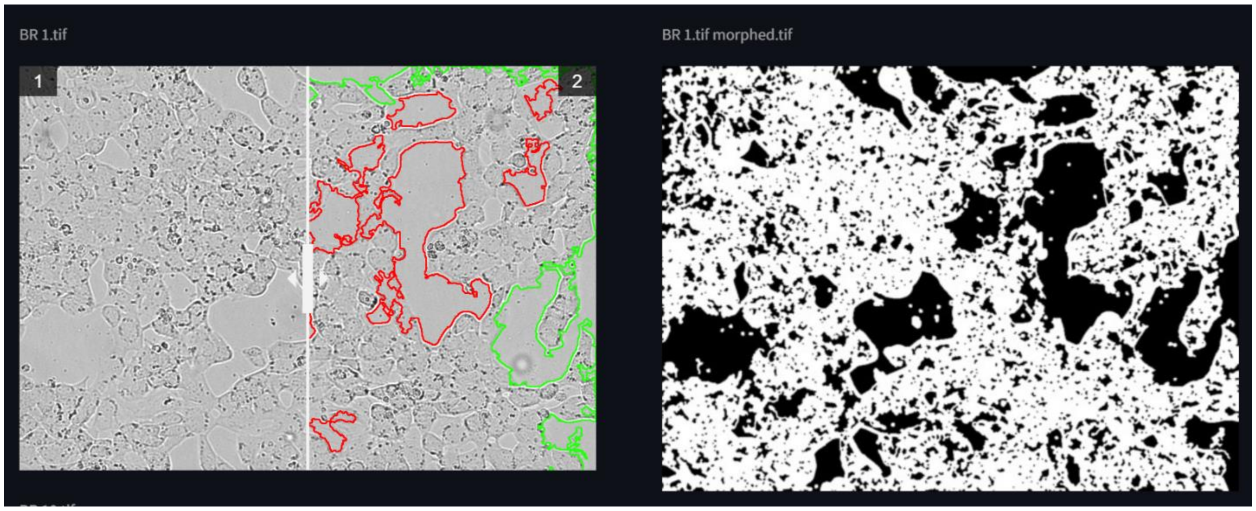
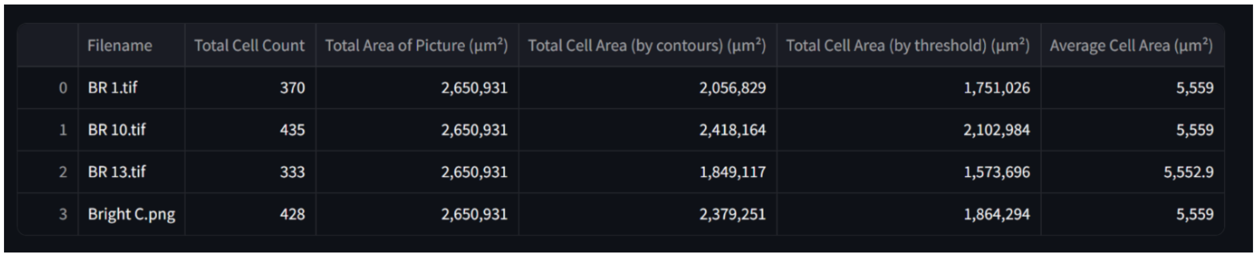


Figure V: Outputted images for single image in batch

Following the images, a table will be shown with all of the data gathered from each image.



	Filename	Total Cell Count	Total Area of Picture (μm^2)	Total Cell Area (by contours) (μm^2)	Total Cell Area (by threshold) (μm^2)	Average Cell Area (μm^2)
0	BR 1.tif	370	2,650,931	2,056,829	1,751,026	5,559
1	BR 10.tif	435	2,650,931	2,418,164	2,102,984	5,559
2	BR 13.tif	333	2,650,931	1,849,117	1,573,696	5,552.9
3	Bright C.png	428	2,650,931	2,379,251	1,864,294	5,559

Figure VI: Table containing all quantitative data retrieved from images

Lastly, the user can download all of the files using the "Download Data" button. This will download a zip file with all overlayed & morphed images, the data in the table, and a settings file for use in the future.