

Fruit detection in ImageJ

A manual covering tasks to detect fruit on plants

Last updated: 12/2/13

Table of contents

What is ImageJ	2
User interface	3
Basic object identification	4
Basic object identification - with color	6
Analyzing particles, counting objects	7
Writing macros	9
Batch processing	11
Plugins	12
Basic image processing glossary	13
Credits	14

What is ImageJ?

ImageJ is an open source, Java-based image processing program.

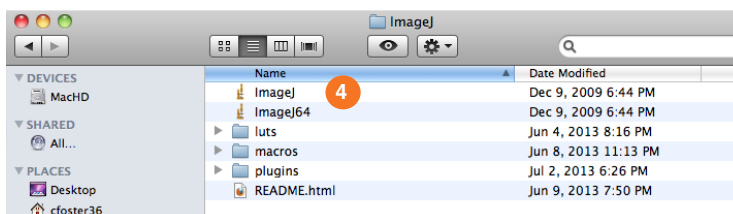
It can be run as an online applet or as a downloadable application.

This instruction manual will focus on the use of ImageJ as an application for finding and counting fruit in images.

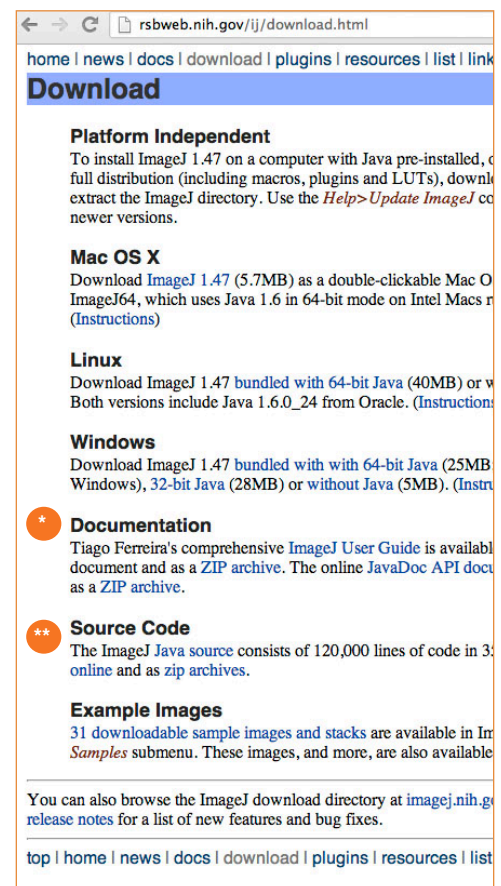


How to download

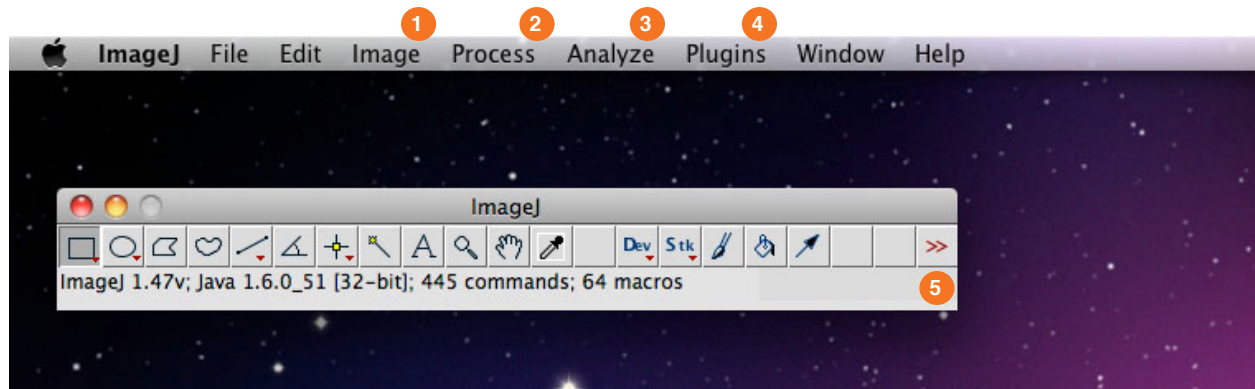
- 1 Go to this url: <http://rsbweb.nih.gov/ij/download.html>
- 2 Click on the appropriate version to download.
- 3 Download and open the ImageJ folder.



- 4 Open the program ImageJ.
- * This is a helpful user guide.
- ** The open source code, if you want to see details or contribute.



User interface



This manual is written using a Mac, but there should be no major differences across systems. ImageJ has a very basic user interface, consisting of the menu and toolbar above.

1

Image

Change the type of image (8-bit, color, etc), flip, rotate, crop, scale, etc. It also includes the “Adjust” menu -- adjusting thresholds, brightness, contrast, etc.

2

Process

Options here involve processing the image, including: binary, find edges, noise, filters, and batch.

3

Analyze

Menu to measure & analyze particles. This will be where fruit is counted.

4

Plugins

Find macros + plugins that you can download and add to the program.

5

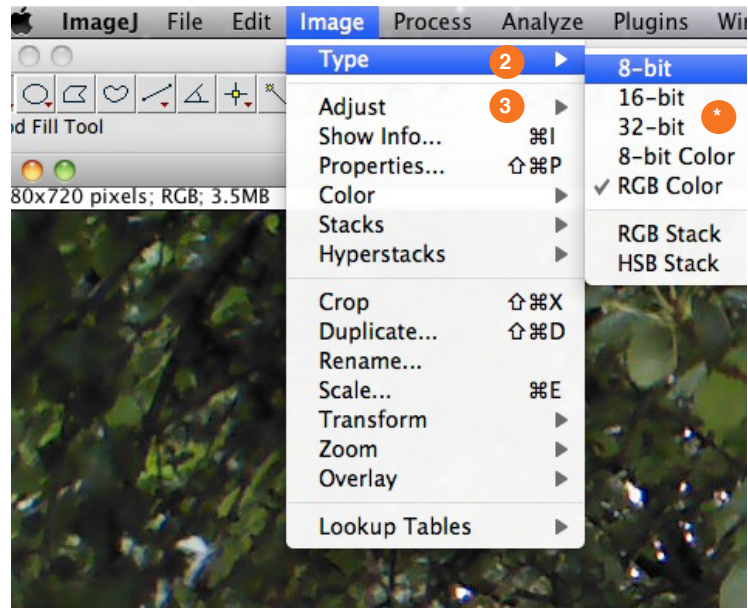
Progress bar

This lists the current progress of any processing/functions.

Basic object identification

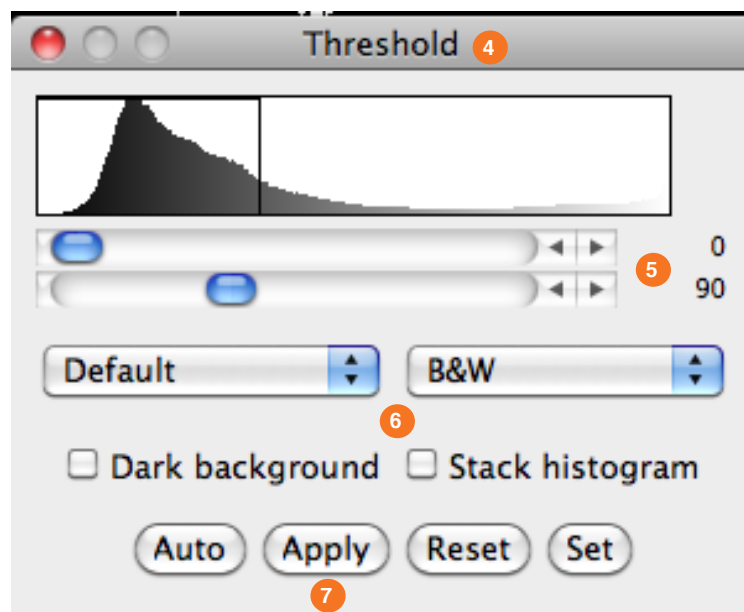
If you have an image where the objects you want to analyze have similar color to the background, like green pears against green leaves, it's best to work with a black and white version. The focus is then on the shape, not the color.

- 1 **Open**
“pears_drone_image_ORIGINAL.jpg”,
or your own file.
- 2 **Convert to grayscale,**
by clicking on Image -->
Type --> 8-bit.
- 3 **Click on Image -->**
Adjust... --> Threshold.



* There is also 16-bit, 32-bit, 8-bit color, and RGB color. This choice depends on what plugins and functions are going to be applied to the image. ImageJ will give you an alert pop-up if you try to do something that does not work in the current type.

- 4 **Adjust the settings for**
the best results.
- 5 Default value is 0. For
this image, 90 produced
the best results.
- 6 The first drop-down
menu should be set at
“default” and the second
at “B&W”.
- 7 **Click apply.**



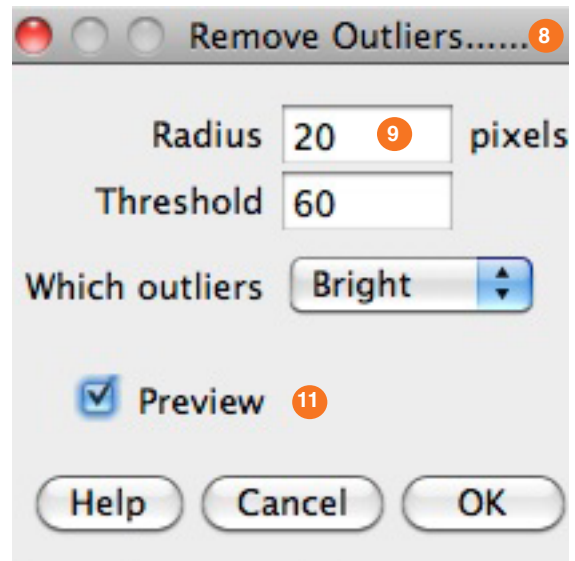
For the pear image, the pears are now visibly circular among the rest of the tree. There is still excess information (white spots, black specks) that we don't want. To get rid of these pixels, we "remove noise".



- 8** Go to Process --> Noise --> Remove Outliers. This window pops up.

These settings are going to be dependent on what gets your photo the best result.

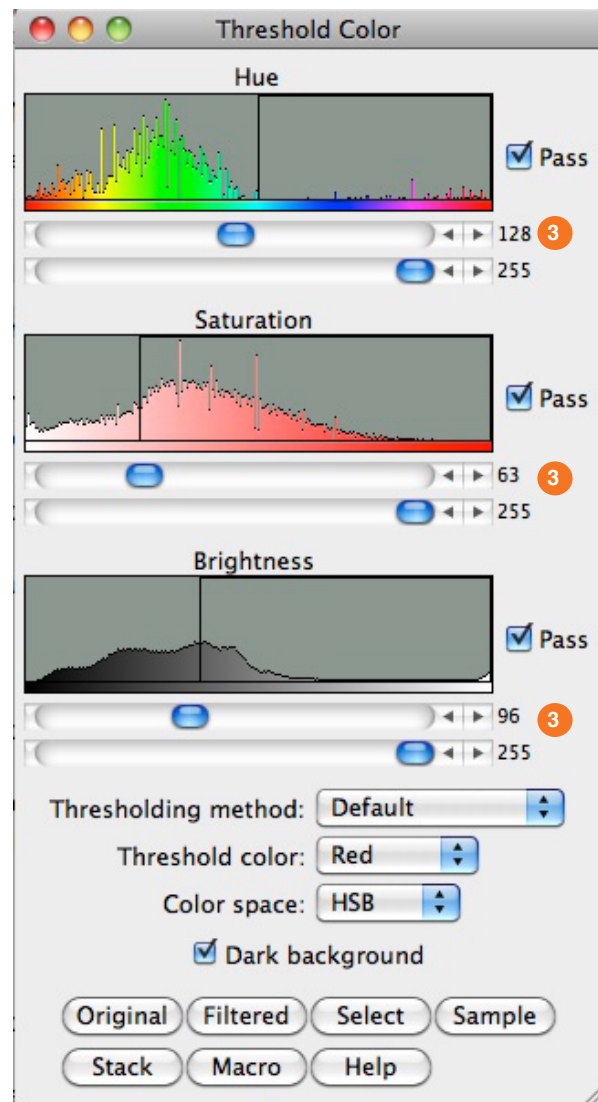
- 9** Anything with a radius smaller than 20 pixels gets eliminated.
- 10** Make appropriate adjustments for the image.
- 11** Checking preview allows you to see the changes in progress.



Basic object identification - with color

If the colors of the objects are distinguishable from the background, or if for some reason you want to work with color, you can do a similar procedure as above involving color.

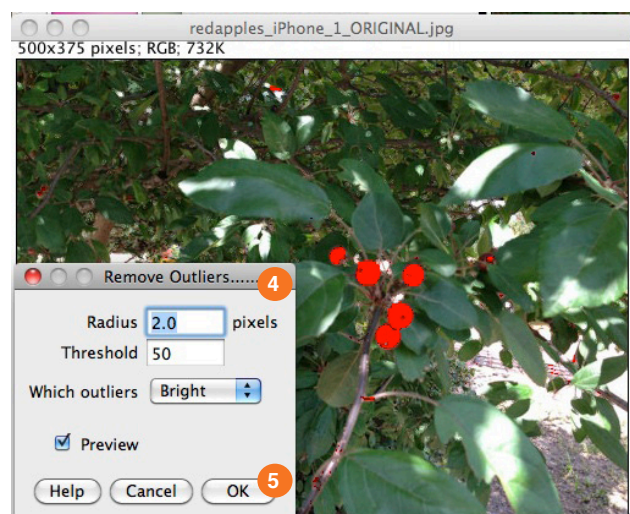
- 1 **Open**
“redapples_iPhone_1_ ORIGINAL”,
or your own file.
- 2 **Go to Image --> Adjust**
--> **Color Threshold.**
This window pops up.
- 3 **Adjust the top bar of**
each setting -- Hue, Sat-
uration, Brightness.



- 4 **Go to Process --> Noise**
--> **Remove Outliers.**
This window pops up.

For my example, I kept the settings at the initial default. Play around with it and see what works best for your image.

- 5 **Click OK.**



Analyzing particles, counting objects

This is a good example to continue with for counting particles. ImageJ was designed by the National Institutes of Health, so it's perfect for counting cells in a microscopic image. We can use this function to count fruit - although its accuracy is questionable.

1 Keep the current image open. Go to Analyze --> Analyze Particles.

2 Adjust the size.

The size depends on the image size. It refers to the number of pixels that make up one shape. 100-Infinity counts particles that are 100 pixels or greater in size.

3 Adjust the circularity.

The circularity is how circular you want the counted objects to be. 0.0 is the least circular -- a straight line. 1.0 is a perfect circle.

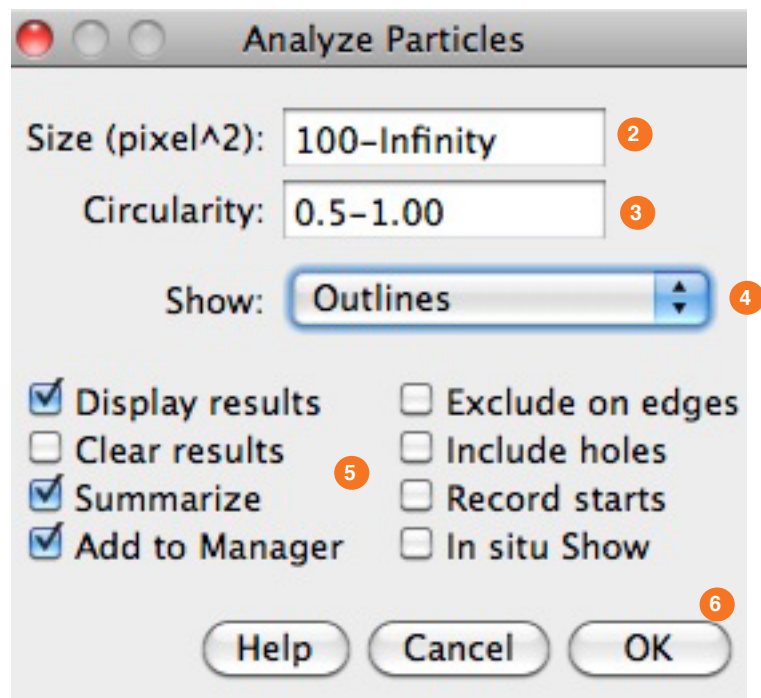
4 Decide what to show.

Outlines generates another image with just the outlines of the counted objects, and the numbers inside of them.

5 Check what you need.

The checkboxes produce different results: "Display results" immediately shows them to you, "Summary" shows a table of results, "Add to Manager" adds them to the ROI (Region of Interest) Manager, "Exclude on edges" means anything counted that intersects with edges of the photo is discarded.

6 Click OK.

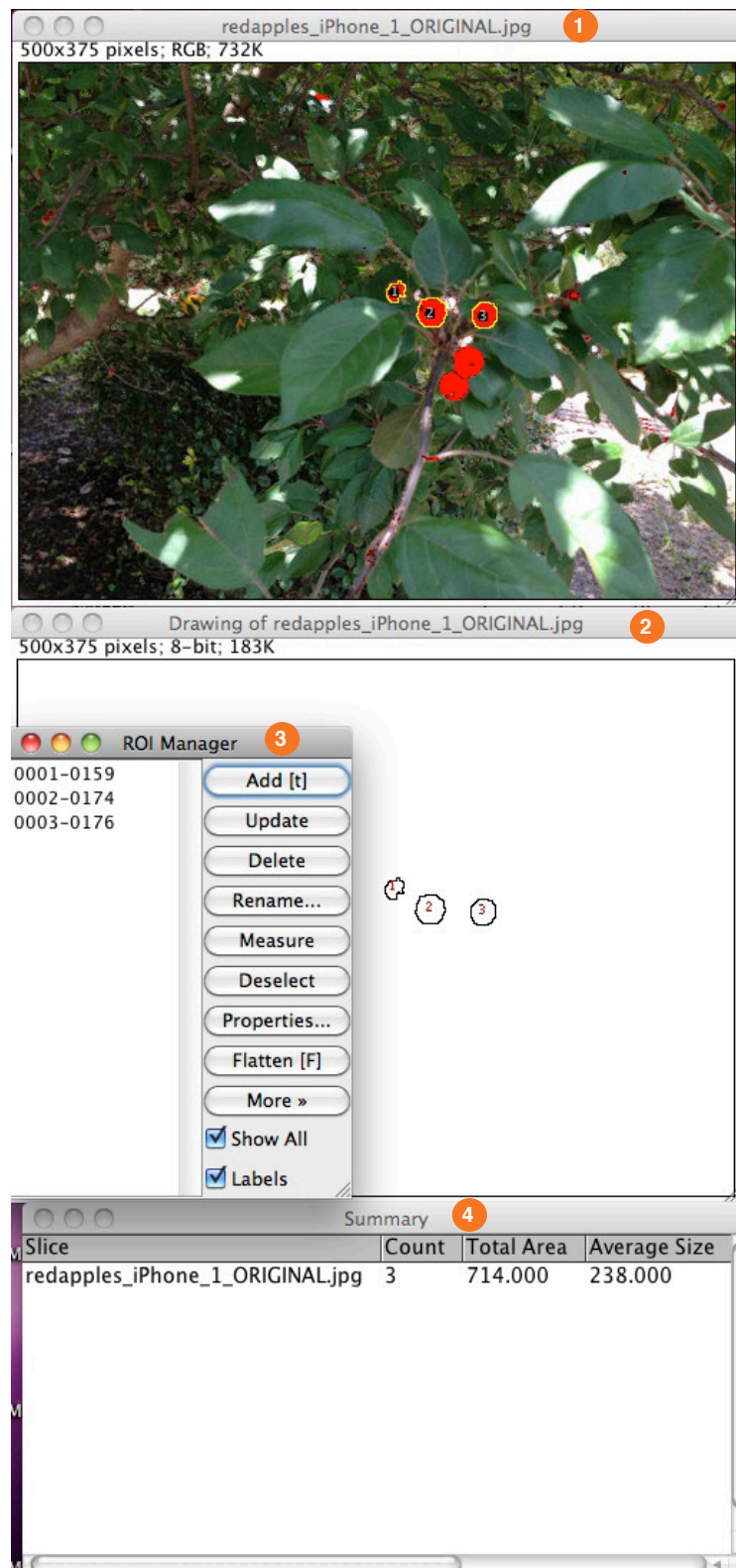


You should get results that look similar to the image on the right. If not, go back through the process and adjust the settings as needed.

- 1 This is the original image with numbered and labeled particles.
- 2 Because I had "Show: Outlines" selected, this window was generated, just showing the outlines of the counted particles.
- 3 Because I had "Add to Manager" checked, the ROI (or Region of Interest) Manager window popped up.

You can open the summary in Excel.

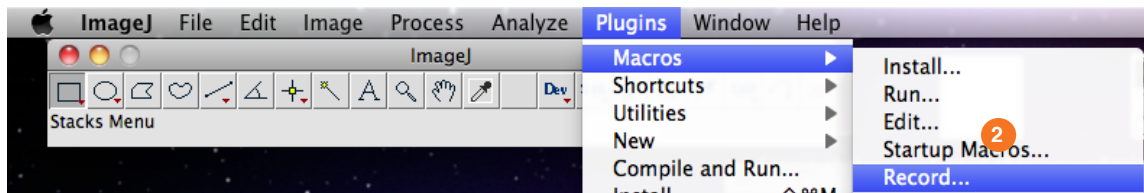
- 4 **Have the Summary box highlighted/active. Go to File --> Save as. Save it on your computer.**



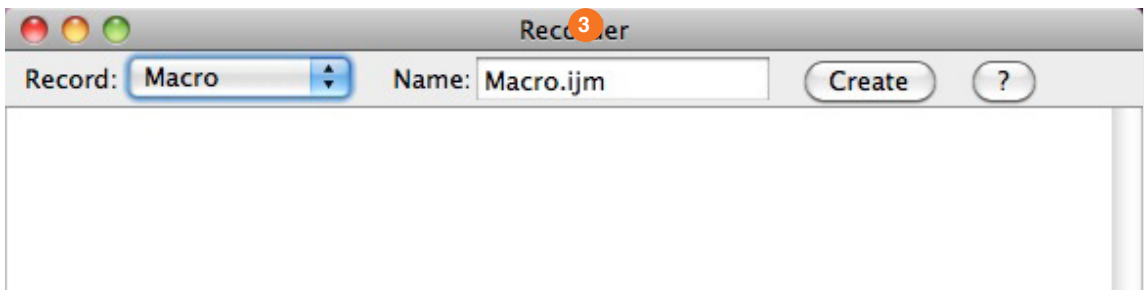
Writing macros

Sometimes, you want to go through the same processes with multiple images. This can become tedious, so there are features to help. You can write a macro and then do batch processing. In ImageJ, a macro is a program of commands that are usually performed manually and individually for one file.

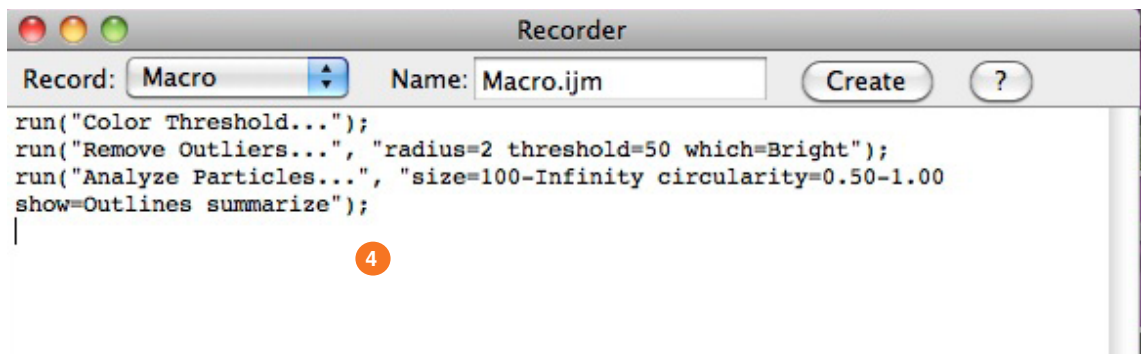
- 1 **Open an image you would like to record the steps for. The examples will be of the previous example.**



- 2 **Go to Plugins --> Macros --> Record...**

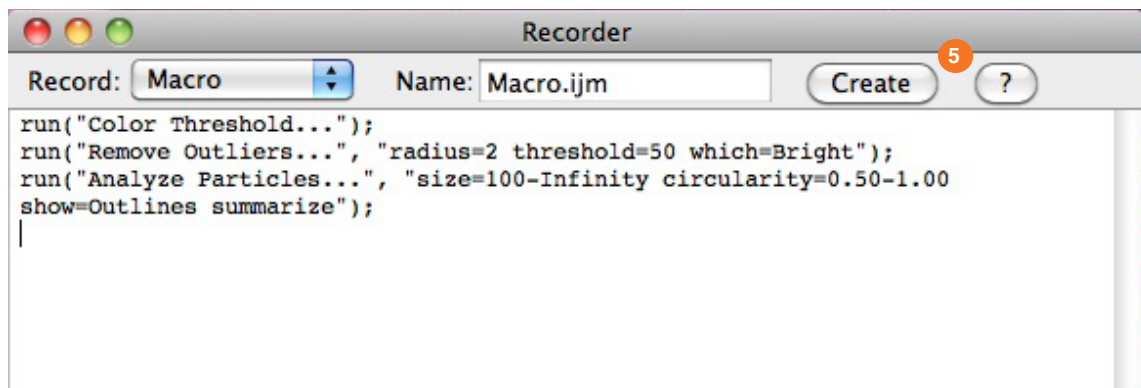


- 3 **The above window should pop up. Name your macro. The file extension is ".ijm"**

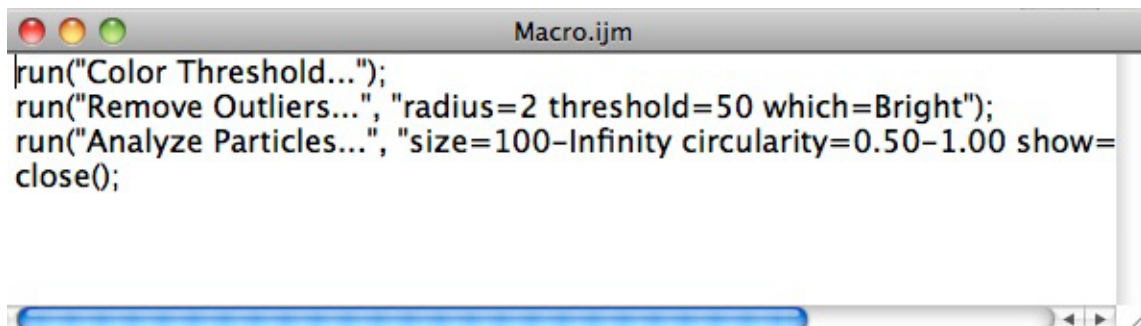


- 4 **Go through the steps you wish to repeat on the other images. As you work, the steps will appear here.**

- ! When you've completed your steps, review the code in the Recorder window. If you closed any window during your process, the line "close();" will be in there. **Delete it.** When the macro is rerun, this line closes **everything**. This almost always causes an error.
- * This demonstrates a handy feature of creating macros – if you know what you want to add, and you're familiar with the code, you can type it in. You can change the generated code, and delete lines if you decide you don't want that in the macro.



- 5 Click "Create" when you are done with your macro. Make sure the name still has ".ijm" at the end. A window like the one below should appear.



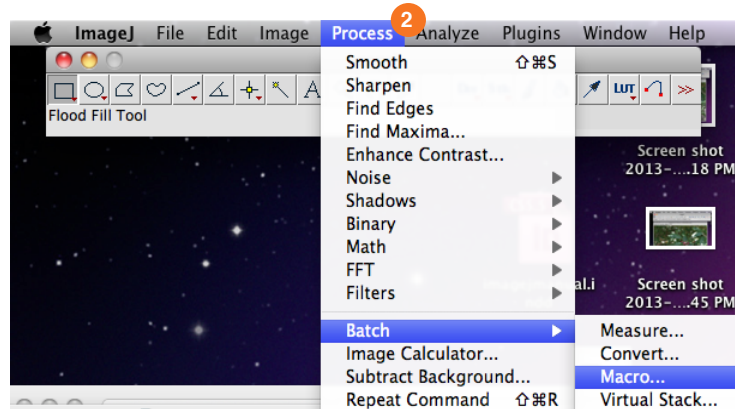
- 6 To use your macro on one image, go to **Plugins --> Macros --> Run.**

A Finder window pops up, and you can locate the macro you just created. It will perform the recorded steps and output the final data.

Batch Processing

What if you want to apply the macro to more than one image, like to a batch of images? This is where macros really come in handy.

- 1 **Open ImageJ. No other files need to be open.**



- 2 **Go to Process / Batch / Macro.**

- 3 **Click “Input” and select the folder of images you want to process.**

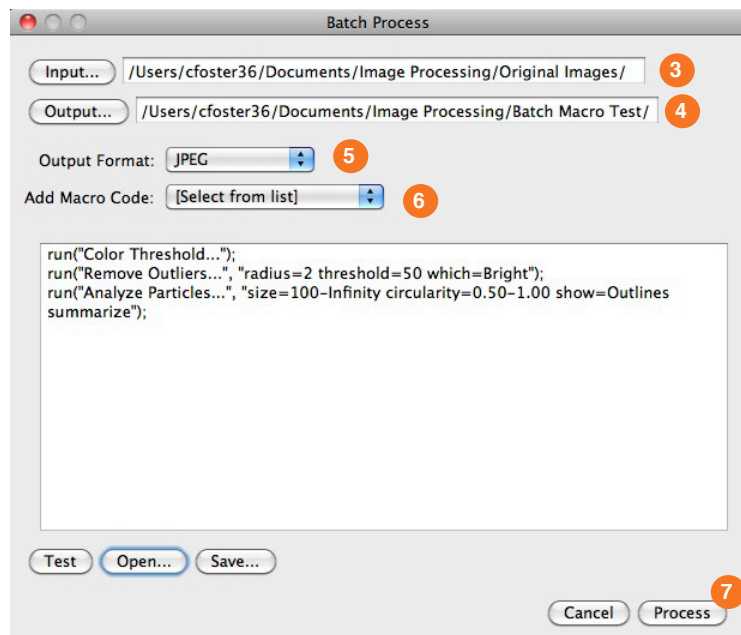
- 4 **Click “Output” and select the folder where you want your process images and results to be saved.**

- 5 **Select the format.**

- 6 **Select the macro.**

- 7 **Click process.**

The data from all of the analyses will be placed in a summary table that will pop up.



Plugins

The initial download of ImageJ has all of the main features. It might be all you need, but if you think of something else you want to do, check to see if there is a plugin for the program. Since it is open source, a lot of people write code to make ImageJ do new things. These usually turn into plugins for anyone to download and use.

1

**Go to <http://rsbweb.nih.gov/ij/plugins/index.html>
Browse and locate the plug-in that does what you are
looking for.**

Hough Circles

Authors: Hemerson Pistori (pistori at ec.ucdb.br)
and Eduardo Rocha Costa (eduardo.rocha at poli.usp.br)

History: 2002/07/29: First version
2006/01/18: Can find circles with arbitrary radius
2008/10/30: Macro compatible
2009/05/04: Works with rectangular selections, thanks to David Webster

Source: [Hough_Circles.java](#)

Installation: Download [Hough_Circles.class](#) to the plugins folder and restart ImageJ.

Description: This plugin applies the Hough Transform for Circles to an 8-Bit image, shows the resulting Hough Space in a new window and marks the centers of the found circles.

The Algorithm performs better if preceded by Equalization + 8-Bit + Edge Detection + Threshold.

The Hough Transform implementation is based on an applet by Mark A. Schulze (<http://www.markschulze.net>)

There will usually be information about the plugin, like the one above. Sometimes all you need to do is click on the source link and drag it to the empty space in the bottom of the ImageJ toolbar. Other times, there are directions, like above, which instruct you to download it and restart it.

2

If you download it, go to Plugins --> Install and find the one you downloaded.

3

Click on it, and it should install.

Depending on the plugin, it might add another tool to the window, or you might access it from the Plugins menu.

4

Restart ImageJ.

Basic Image Processing Glossary

This is a section describing basic image processing and functions. These terms apply anywhere with analyzing images – not just ImageJ.

Pixel

Pixels are the foundation of any image – you can think of them as small dots that make up a larger image. A pixel can only be one color, but for an average image there are enough the pixels and colors to blend together well, where each pixel is indistinguishable.

8-bit color

Without getting into too much math, 8-bit allows for 2^8 , or 256, colors to be displayed. 16 and 32-bit color allows for more colors to be displayed, 2^{16} and 2^{32} .

RGB color

RGB stands for Red Green Blue. These are three hues of light that make up color in digital images. Many applications involve adjusting each hue, so it's important to understand. Each hue has a value from 0 to 255. If all 3 are set to 255, it is black, and if all 3 are set to 0, it is white.

Threshold

A threshold is a simple image segmentation method. Typically a cutoff point is selected. This is a number where values below it become a certain color, and values above become another. It works for RGB and grayscale images. It's frequently used for object identification in images

Noise

Noise is random variation in brightness or color information in images, according to Wikipedia. In image processing, it is usually something you want to minimize because it can make object selection and detection harder.

Credits

ISTC • SOCIAL



People involved in this project are:

Carl DiSalvo, Research Project Lead

Caroline Foster, Designer, ceefoster2@gmail.com

Tasmia Alam, Designer, tasmiaalam3@gmail.com