Image Analysis using MATLAB GUI

Xiaoquan Sun

project.m is a MATLAB GUI designed to measure the fluorescent intensity of plant tissue images acquired from the fluorescent microscopy, to count the number of vacuoles in the selected region and to calculate the area of all the vacuoles.

Instructions (Video Demo: https://goo.gl/yywrij)

`run project.m` to start the MATLAB GUI

1. **Load Image** can load images and display it in display region.

2. **Reset Image** can reload the previous image file and display it in display region.

3. **Choose Region** can choose a rectangle region in display region. You may rotate the image by tuning **Rotate Image bar** to choose a best angle for the selection. **Right click** and choose **Crop image**, the selected region will be shown in the display region.

4. **Florescence Intensity** will give an averaged intensity of the selected region. The darkest is 0, and the brightest is 1. STD (Standard Deviation) is also given. Results will show in the output region.

5. **Counting** will give the counting results. Vacuoles will be circled with green line. And ones labeled with red circled will be counted and shown in the output region. The minimum and maximum of vacuoles area will be shown in corresponding region. You may change them manually and **Modified Counting**, new output will be given.

6. The distribution will give the distribution of vacuoles automatically. The x axis is vacuole area, unit is pixel, and the y axis is the number of vacuoles.

7. If the counting is not good, **+** can be used to modify the figure. By drawing black lines to separate the connected vacuoles, then press **Counting** or **Modified Counting**, new result will be given.

8. **-** can draw a white line. Connect the vacuoles you want to delete with a vacuole cut by the boundary, then press **Counting** or **Modified Counting**, the selected vacuoles will be removed.

1. Digital Image Data Format

The most important thing before building the MATLAB GUI is to understand what is an image. The most often image file formats are PNG, JPG, GIF and TIF. These digital images are rasterized to a grid of pixels. Pixels have two coordinates (x, y) that indicate the location on the image, and also contain a set of data of the color information. The color information is called the color depth. Primary colors are stored as binary format with 0 and 1 and measured as bits per pixel in the color depth. More bits will create more colors, and the image will be better. A 1 bit image only had $2^1 = 2$ colors, which is black (0) and white (1). An 8 bit image has $2^8 = 256$ colors, which contains 3 bit red, 3 bit green and 2 bit blue. For a 24 bit JPG image which is also called TrueColor image, each primary color has 8 bit data. And the image can present $2^{24} = 16,777,216$ colors. Besides primary colors, color depth also contains information about transparency or gray scale for different condition.

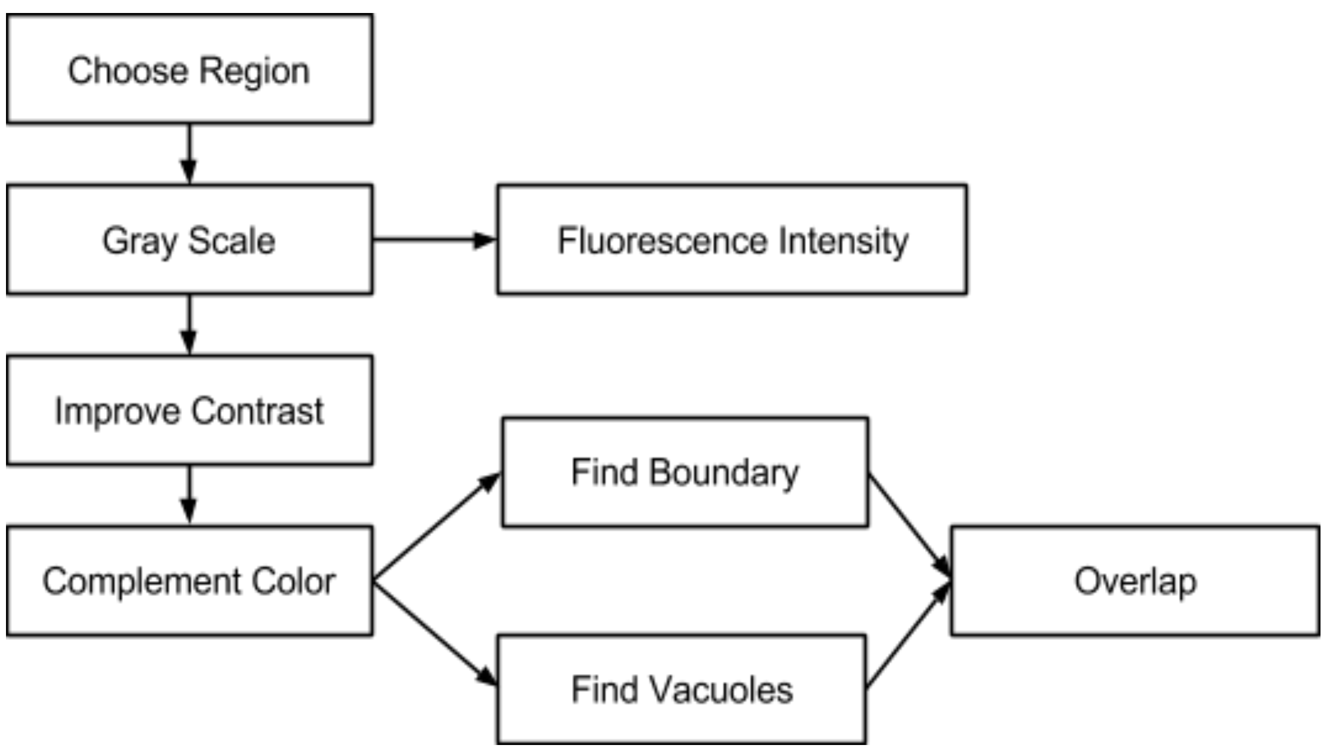

Figure 1. The workflow in MATLAB GUI.

2. MATLAB GUI Script

Image Processing Toolbox in MATLAB has many powerful commands to analyze or modify image files. In this MATLAB GUI, the image is handled in the following procedures:

1. Choose the area of interest;
2. Convert it to gray scale image to analyze fluorescence intensity;
3. Improve the contrast of the gray scale image;
4. Find the complementary colors and convert the new image to binary image;
5. Find the vacuoles and their boundaries;
6. Overlap the Vacuoles and boundaries and output the analysis.

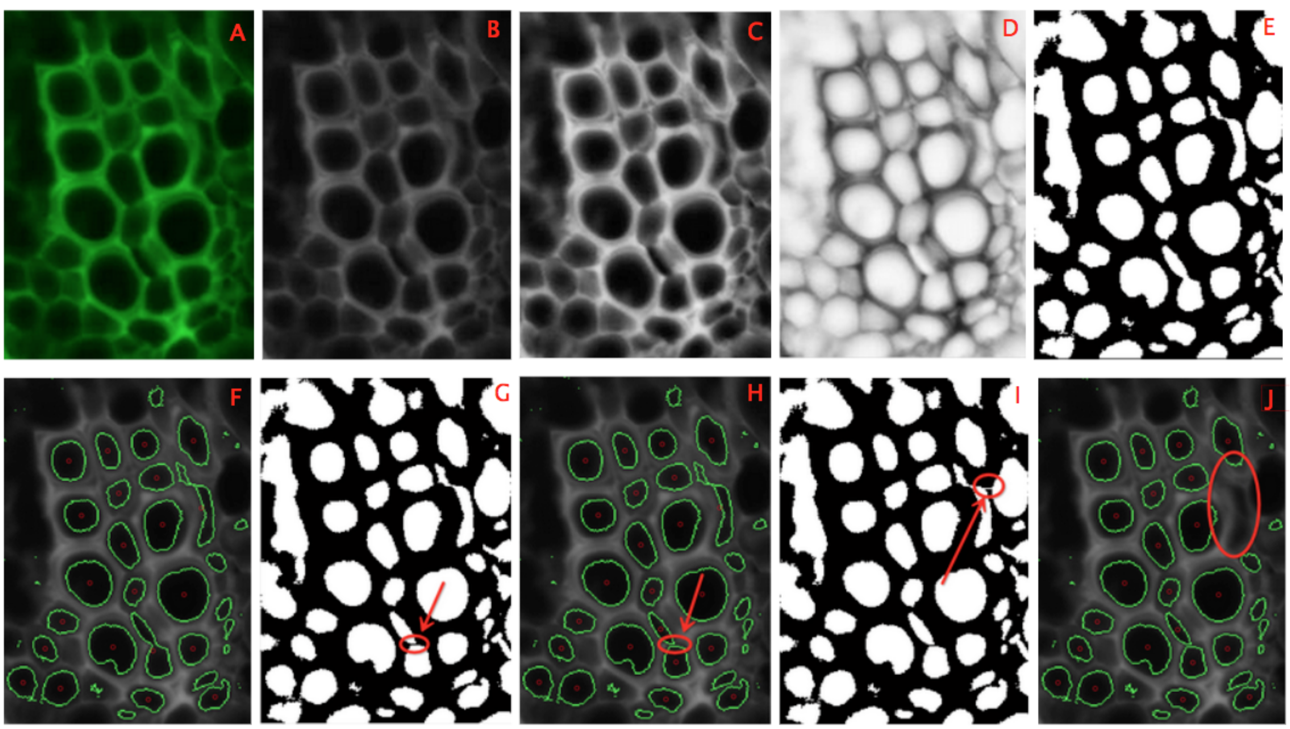

Figure 2. Illustration of the whole procedure. A: original image; B: gray scale image; C: contrast-improved image; D: complementary color image; E: binary image; F: primary analysis result; G: add the black line to separate connected vacuoles; H: modified result of G; I: add the white line to link objects with boundary to delete them; J: modified result of I.

Commands used in this MATLAB GUI script include ***imread, rgb2gray, im2bw, imcomplement, adaphtisteq, imfill, imclearborder, bwconncomp, regionprops***. They are the cores of the GUI. The following is a brief introduction of each of them.

imread

MATLAB uses “imread” to import the image data. The format of the image file is determined by the file extension. The image will be converted to a 3D array which can be represented as X*Y*N. X and Y are the plane coordinates. N is called colormap in MATLAB; it is the color depth and is used to store the information of the primary color.

rgb2gray

“rgb2gray” can convert a colorful image to a gray scale image. The gray scale image only has luminance intensity. “rgb2gray” converts RGB values to gray scale values by sum of the R, G, and B: 0.2989*R + 0.5870*G + 0.1140*B, each primary color has respective weight. Gray scale image can be used to compare the fluorescence intensity. The averaged intensity of selected region and the standard deviation are calculated in the GUI.

im2bw

“im2bw” converts a gray scale image to a binary image. A threshold is set to distinguish white and black. The intensity value is between 0 and 1. The pixel has intensity greater than the threshold will be 1 (white), and the rest are all 0 (black). The default threshold is 0.5. In this project, the threshold is set between 0.5 and 0.8 to get the best effect.

imcomplement

“imcomplement” can compute the complementary color of each pixel. In a B&W image, 0 is converted to 1, and 1 is converted to 0. The process is not essential. But in this project, to make sure MATLAB can deal with the figure for the segmentation; vacuoles will be represented as white holes by this command.

adaphtisteq

“adaphtisteq” is used to enhance the contrast of the gray scale image by transforming the values using contrast-limited adaptive histogram equalization (CLAHE). CLAHE was designed to improve the contract for biology images. Due to the great effect, it is well used in image process field now. In this method, the entire image is divided into many small regions. Each region contract is enhanced individually. The algorithm to enhance contrast is equalizing the histogram. The equalized region was then combine by bilinear interpolation. Since CLAHE operates on small regions in the image instead of the whole image. The method can avoid overamplification of noise.

imfill

The algorithm of “imfill” is from morphological reconstruction. In this GUI, “imfill” fills hoses in vacuoles. And the boundary of these holes (which is the boundary of vacuoles) can be found and shown as green lines to highlight the selection.

bwperim

This command outputs the perimeter pixels of objects (white holes in this case) in the input image. Combined with imfill. The boundary of vacuoles can be found.

imclearborder

The algorithm of “imclearborder” is based on morphological reconstruction. This command can find the vacuoles which connecte by the boundary of the figure and expel them in the following analysis. In the GUI, this command is also used to delete the vacuoles that we are not interested in by connecting the object with boundary with a white line.

bwconncomp, regionprops

This command “bwconncomp” is used to segment vacuoles. MATLAB will first search a pixel that is not be searched before, and then it uses a Flood-Fill algorithm to find the connected pixels around the pixel. The result is a vacuole. By looping this process, all vacuoles can be found. A manual modification tools is added in the GUI based on this command. For those objects that connect with each other at one boundary, one can draw black line at the boundary to separate them. After this, “regionprops” will be used to get the area and coordinates of center of mass of each vacuole. By defining an interested area range, the selected vacuoles for further analysis will be label a red circle at the center of mass.

3. **Watershed Algorithm**

Watershed algorithm is used in this GUI to segment vacuoles. It is a simple method to find the connected regions from a reference region. The region connect with the reference region in different directions will be counted as one big region. The detection of neighboring regions has two types: 4- and 8-Connected Neighborhoods. In 4-connected neighboring, the region shares the same side with the reference region will be counted. In 8-connected neighboring, 4 more diagonal regions will be counted. In this GUI, 8-connected neighboring is used.

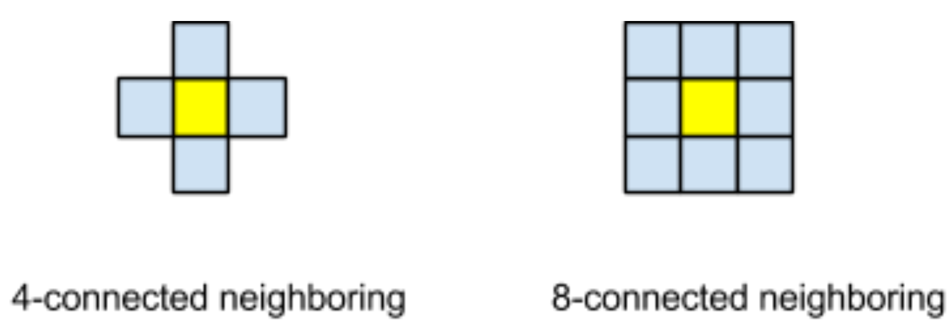

Figure 3. Illustration of 4, 8-connected neighboring