# **Finding Parasitic Infections with MATLAB**

**Functions Used:**

imgSet = imageSet(imageLocation)

returns an object for storing an image data set or a collection of image data sets.

- contains image descriptions, locations of images, and the number of images in your collection.

toggle fig:

Finds and activates, or creates, figures with user-specified names.

If no name is provided, creates figure named "untitledn"

H = gobjects(s1,...,sn) returns an s1-by-...-by-sn graphics object array,

where the list of integers s1,...,sn defines the dimensions of the array.

For example, gobjects(2,3) returns a 2-by-3 array.

subplot(m,n,p) divides the current figure into an m-by-n grid and

creates axes in the position specified by p.

[pathstr, name, ext, versn] = fileparts('filename') returns the path, filename, extension, and version for the specified file. The returned ext field contains a dot (.) before the file extension.

xticks([ticks](https://in.mathworks.com/help/matlab/ref/xticks.html#bu6jwmu-ticks)) sets the *x*-axis [tick values](https://in.mathworks.com/help/matlab/ref/xticks.html#bvaecsy-7), which are the locations along the *x*-axis where the tick marks appear. Specify ticks as a vector of increasing values; for example, [0 2 4 6]. This command affects the current axes.

[BW2](https://in.mathworks.com/help/images/ref/bwareaopen.html#buwet8w-1-BW2) = bwareaopen([BW](https://in.mathworks.com/help/images/ref/bwareaopen.html#buwet8w-1-BW),[P](https://in.mathworks.com/help/images/ref/bwareaopen.html#buwet8w-1-P),[conn](https://in.mathworks.com/help/images/ref/bwareaopen.html#buwet8w-1-conn)) removes all connected components, where conn specifies the desired connectivity.

**drawnow** (**MATLAB** Functions) **drawnow** flushes the event queue and updates the figure window. as an M-file updates the current figure after executing the **drawnow function** and after executing the final statement

Bwmorph

'skel'

With n = Inf, removes pixels on the boundaries of objects but does not allow objects to break apart. The pixels remaining make up the image skeleton. This option preserves the Euler number.

When working with 3-D volumes, or when you want to prune a skeleton, use the bwskel function.

'spur'

Removes spur pixels. For example:

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 1 0 becomes 0 0 0 0

0 1 0 0 0 1 0 0

1 1 0 0 1 1 0 0

Create a disk-shaped structuring element. Use a disk structuring element to preserve the circular nature of the object. Specify a radius of 10 pixels so that the largest gap gets filled.

se = strel('disk',10);

Perform a morphological close operation on the image.

closeBW = imclose(originalBW,se);

[BW2](https://in.mathworks.com/help/images/ref/bwpropfilt.html#bufeg5j-1-BW2) = bwpropfilt([BW](https://in.mathworks.com/help/images/ref/bwpropfilt.html#bufeg5j-1-BW),[attrib](https://in.mathworks.com/help/images/ref/bwpropfilt.html#bufeg5j-1-attrib),[range](https://in.mathworks.com/help/images/ref/bwpropfilt.html#mw_dfb70b78-b872-4e33-be13-9f76a9a5760d)) extracts all connected components (objects) from a binary image BW whose value of property attrib is in the specified range. bwpropfilt returns a binary image BW2 containing only those objects that meet the criteria.

Test for infection:

imhistmatch([I](https://in.mathworks.com/help/images/ref/imhistmatch.html#bth_8q2-A),[ref](https://in.mathworks.com/help/images/ref/imhistmatch.html#bth_8q2-ref)) transforms the 2-D grayscale or truecolor image I returning output image J whose histogram approximately matches the histogram of the reference image ref.

[[X](https://in.mathworks.com/help/matlab/ref/meshgrid.html#bvblp7l-X),[Y](https://in.mathworks.com/help/matlab/ref/meshgrid.html#bvblp7l-Y)] = meshgrid([x](https://in.mathworks.com/help/matlab/ref/meshgrid.html#bvblp7l-x),[y](https://in.mathworks.com/help/matlab/ref/meshgrid.html#bvblp7l-y)) returns 2-D grid coordinates based on the coordinates contained in vectors x and y.

Any : Determine if any array elements are nonzero

Show mask as overlay:

varargin is an input variable in a function definition statement that enables the function to accept any number of input arguments.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Brief about functions used in code -2**

**HSV or Hue Saturation Value** is used to separate image luminance from color information.

'Hue' represents the color

'Saturation' represents the amount to which that respective color is mixed with white

'Value' represents the amount to which that respective color is mixed with black (Gray level).

Using this model, we can detect an object with a certain color

to reduce the influence of light intensity from the outside.

Histogram is a graphical representation of the intensity distribution -

it represents the number of pixels for each intensity value considered.

**Histogram equalization** maps the grey scale such that the output image uses the

entire range available and such that there are approximately the same number of pixels of

each grey value in the output image

This allows for areas of lower local contrast to gain a higher contrast.

Logical operations are done on pixel by pixel basis.

The **AND and OR** operations are used to isolate the interested region from the rest of the image portion

This masking operation is referred to as Region Of Interest processing.

Consider a mask image L for the image A.

To obtain the interested area, D= and(L,A) ;

We can use L&A also.

The resulting image will be stored in D which contains the isolated image part.

**Image masking** is a process to hide some portions of an image and to reveal some portions.

It is a non-destructive process of image editing.

Masking involves setting some of the pixel values in an image to zero, or some other "background" value.

A mask image is simply an image where some of the pixel intensity values are zero, and others are non-zero.

**Canny Edge Detection** is a popular edge detection algorithm, consists of :

1 Noise Reduction

2 Finding Intensity Gradient of the Image

3 Non-maximum Suppression

4 Hysteresis Thresholding

**bwareaopen:**

Remove objects containing fewer than ‘n’ pixels

**Bwlabel -** takes in a binary image.

Pixels that belong to an object are denoted with 1 / true while those pixels that are the background are 0 / false.

For example:

0 0 0 0 0 1 1 1 0 0

0 1 0 1 0 0 1 1 0 0

0 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0 1 1

0 0 1 1 1 1 0 0 1 1

The output of bwlabel will give you an integer map where each object is assigned a unique ID.

0 0 0 0 0 3 3 3 0 0

0 1 0 1 0 0 3 3 0 0

0 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 4

0 0 0 0 0 0 0 0 4 4

0 0 2 2 2 2 0 0 4 4

**Regionprops :**  measures a variety of image quantities and features in a black and white image.

One of these particular properties is the centroid.

s = regionprops(img, 'Centroid');

The above code will calculate the centroids of each of your objects in the image.

%Store the x- and y-coordinates of the centroids into a two-column matrix.

disp(cat(1,s.Centroid))

3.0000 2.6000

4.5000 6.0000

7.2000 1.4000

9.6000 5.2000

As such, the first centroid is located at (x,y) = (3, 2.6)