

# Basic Image Processing with MATLAB Programming

This live script includes information on how to read images, what the read images contain, and simple image manipulation.

## Table of Contents

Load an image.....	1
Display the image.....	2
Brightening/darkening Images.....	2
Resizing images.....	3
Convert to Grayscale.....	4
Segmenting an image.....	5

First, clear the workspace and start new.

```
close all
clearvars
clc
```

## Load an image

Use the 'imread' command to load up images.

```
% Read images using imread. Change the path for your own images.
I = imread('peppers.png');

% What does the following line do?
I(1:5, 1:5, 1)
```

```
ans = 5x5 uint8 matrix
    62    63    63    65    66
    63    61    59    64    63
    65    63    63    66    66
    63    67    67    63    64
    63    62    64    65    66
```

```
I(1:5, 1:5, 2)
```

```
ans = 5x5 uint8 matrix
    29    31    34    30    27
    31    31    32    30    28
    29    30    31    30    31
    29    29    31    31    31
    31    32    33    30    31
```

```
I(1:5, 1:5, 3)
```

```
ans = 5x5 uint8 matrix
    64    64    64    60    59
    62    64    64    60    59
    60    62    63    61    61
    62    63    63    60    62
    62    63    62    61    61
```

You can see that an image is nothing more than a 3-dimensional matrix of pixel colour intensities. The 1st dimension is the number of rows (height), the 2nd dimension is the number of columns (width) and the 3rd dimension is the colour channel (Red/Green/Blue or RGB).

Each pixel has an RGB colour intensity from 0-255, with 0 being black and 255 being completely bright.

## Display the image

```
imshow(I)
```



## Brightening/darkening Images

Since an image is just a matrix of numbers, we can manipulate it mathematically. Note that higher numerical values indicates brighter pixels, and lower numbers are darker pixels.

```
I_bright = I*1.25; % ~25% brighter  
imshow(I_bright)
```



```
I_dark = I*0.75; % ~25% darker  
imshow(I_dark)
```



**Resizing images**

```
I_resized = imresize(I,[100 100]); %resize to 100x100
imshow(I_resized)
```



Selecting portions of an image is like selecting rows and columns of a matrix:

```
topQuarter = I(1:end/2,1:end/2, :); % What is the ':' doing at the end here?
imshow(topQuarter)
```



## Convert to Grayscale

The image you see as is, is colored. The color information is encoded as a RGB triplet - the first 'channel' includes the Red information, second includes Green, and third has Blue. Sometimes, you may want to compress all that information into a single channel, and look at the objects in a grayscale format. Here's how you do it:

```
Igray = rgb2gray(I);
imshow(Igray)
```

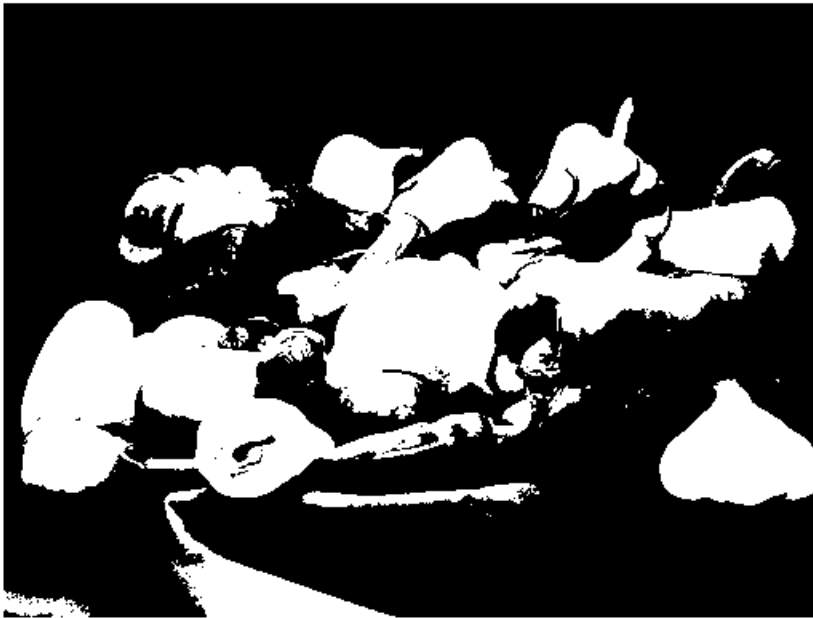


## Segmenting an image

Segmentation is a process of separating parts of interest in an image -

In a grayscale image, if we set all dark values, say over 100 to 255 and all dark value (below 100) to 0, we can get a binary boundary separating the light part of the image from the dark part.

```
Iseg = Igray;  
Iseg(Iseg>=100) = 255; % retain bright parts  
Iseg(Iseg<100) = 0;   % remove dark parts  
  
imshow(Iseg)
```



We can do this for the colour image as well - leading to some strange results. Can you explain the strangeness?

```
Iseg = I;  
Iseg(Iseg>=100) = 255; % retain bright parts  
Iseg(Iseg<100) = 0;    % remove dark parts  
  
imshow(Iseg)
```

