Lecture 9: Image Processing in MATLAB

Image Processing in MATLAB provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development. You can perform image segmentation, image enhancement, noise reduction, geometric transformations, and image registration using deep learning and traditional image processing techniques. It supports processing of 2D, 3D, and large images.

1. Finding image information

Information about graphics file:
To get information about an image:
```
info=imfinfo('Drive:/Filename.Format');
```

Image name:
```
image_name=info.Filename
```

The last modification date:
```
modification_date=info.FileModDate
```

To know the image size:
```
image_size=info.FileSize
```

Width of the image:
```
image_width=info.Width
```

Height of the image:
```
image_height=info.Height
```

To know the color depth of a picture:
```
color_depth=info.BitDepth
```

To define the color type:
```
color_type=info.ColorType
```

To find the number of samples:
```
sample_number=info.NumberOfSample
```

Coding method:
```
coding_method=info.CodingMethod
```

Coding process:

```
coding_process=info.CodingProcess
```

…

Example: Find the information about an image located on a locale drive in your PC.

Ans.: `info=imfinfo('C:/Dog.jpg');`

2. Reading, Changing Format, and Writing Image

To read an image changes its format and store it back in the local location using MATLAB.
To do that, let's take a matrix name image and to load an image into this matrix, we have to use a command called `imread`. This function takes the location of the image as argument followed by the name of the image and format of the image.
To change the format of the image and write it back to the local storage, we have to use another command called `imwrite`.

Example: Read an image located on the local storage change the format and save it back with a different name.

Ans.:

```
image=imread('C:/Dog.jpg');
imwrite(image,'C:/newDog.png','png');
```

3. Displaying images in MATLAB

```
image=imread('C:/image name.format');
imshow(image);
```

To display more images in one window in MATLAB, we need to read all the images we would like to display then using the command `subplot` to open the images on one window.

```
image1=('C:/image name.format');
image2=('C:/image name.format');
image3=('C:/image name.format');
image4=('C:/image name.format');
subplot(4,1,1);imshow(image1);
subplot(4,1,2);imshow(image2);
subplot(4,1,3);imshow(image3);
subplot(4,1,4);imshow(image4);
```

Example: Read four images located on the local storage and display all the images on one window.

```
image1=('C:/Dog.jpg');
image2=('C:/Dog1.jfif');
image3=('C:/Dog2.jfif');
```

```
image4=('C:/Dog3.jpg');
subplot(4,1,1);imshow(image1);
subplot(4,1,2);imshow(image2);
subplot(4,1,3);imshow(image3);
subplot(4,1,4);imshow(image4);
```

4. Change Image Color Space using MATLAB

To change an image space, we may use `rgb2gray` command in which, the image color change to gray.
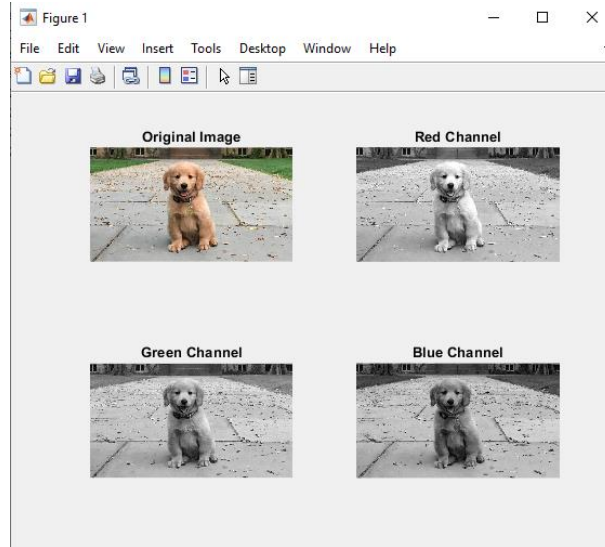
Example: Read the Dog image located on C:/ and display the original and gray image on one window.

```
original_image=imread('C:/Dog.jpg');
gray_image=rgb2gray(original_image);
subplot(2,1,1);imshow(original_image);
subplot(2,1,2);imshow(gray_image);
```

5. Splitting RGB Channels from an Image

Example1: Read the Dog image located on C:/, splitting RGB channels from the image and display the original and all channels on one window.

```
image=imread('C:/Dog.jpg');
red_channel=image(:,:,1);
green_channel=image(:,:,2);
blue_channel=image(:,:,3);
subplot(2,2,1);imshow(image);title('Original Image');
subplot(2,2,2);imshow(red_channel);title('Red Channel');
subplot(2,2,3);imshow(green_channel);title('Green Channel');
subplot(2,2,4);imshow(blue_channel);title('Blue Channel');
```

Example2: Read the Color image located on C:/, splitting RGB channels from the image and display the original and all channels on one window.

```matlab
image=imread('C:/Color.jpg');
red_channel=image(:,:,1);
green_channel=image(:,:,2);
blue_channel=image(:,:,3);
subplot(2,2,1);imshow(image);title('Original Image');
subplot(2,2,2);imshow(red_channel);title('Red Channel');
subplot(2,2,3);imshow(green_channel);title('Green Channel');
subplot(2,2,4);imshow(blue_channel);title('Blue Channel');
```