Student: Melis Kilic

Introduction to Digital Image Processing and Matlab software

I-1 Basic Matlab operations

a) Create a vector A=[0, 10, 20, 30, …300];

```
A = 0 : 10 : 300
```

*b) Create a column 10-element vector: B=[5 5 5 5 …. 5]T. Use: …*ones(…, …)+…*

```
B = 5*ones(10,1);
```

c) Create a matrix C=2x6 with all zeros. Use Matlab function zeros.

```
C = zeros(2,6);
```

*d) Create a matrix D 4x6 with random elements between 5 and 10. Use: …*rand(…,…)+…*

```
D = 5 + (10-5) * rand(4, 6);
```

e) Assign E=D; and then replace rows 2 and 3 of matrix E with matrix C

```
E = D;
E(2,:)= C(1,:);
E(3,:)= C(1,:);
```

f) Create a matrix F:

```
F = [ones(8,1) [D; E] ones(8,1)];
```

I-2 Displaying images

a) Display the image lena.jpg. Use function imread to read the image to matrix A and then imshow to display the matrix (see help for details).

```
figure;

subplot(2,2,1);
A = "lena.jpg";
imshow(A);
title('Original image');
```

b) Convert the image to grayscale using rgb2gray (assign the grayscale image to new variable B) and display it. Before you display new image you may use figure to open new window, so the previous image will be preserved.

```
subplot(2,2,2);
B = imread("lena.jpg");
B = rgb2gray(B);
imshow(B);
title('Grayscale image');
```

c) Greyscale images can be treated as a two-dimensional brightness function f(x,y), where: x, y are the coordinates of the pixel and f(x, y) is the brightness level (usually 0 – 255). Display image lena.bmp as a two-dimensional function using Matlab function mesh. It is necessary to convert the data into a double type – e.g. mesh(double(A)). Display the result. Rotate the image using the mouse.

```
subplot(2,2,3);
C = imread("lena.bmp");
C = rgb2gray(C);
mesh(double(C));
title('Mesh f(x,y)');
```

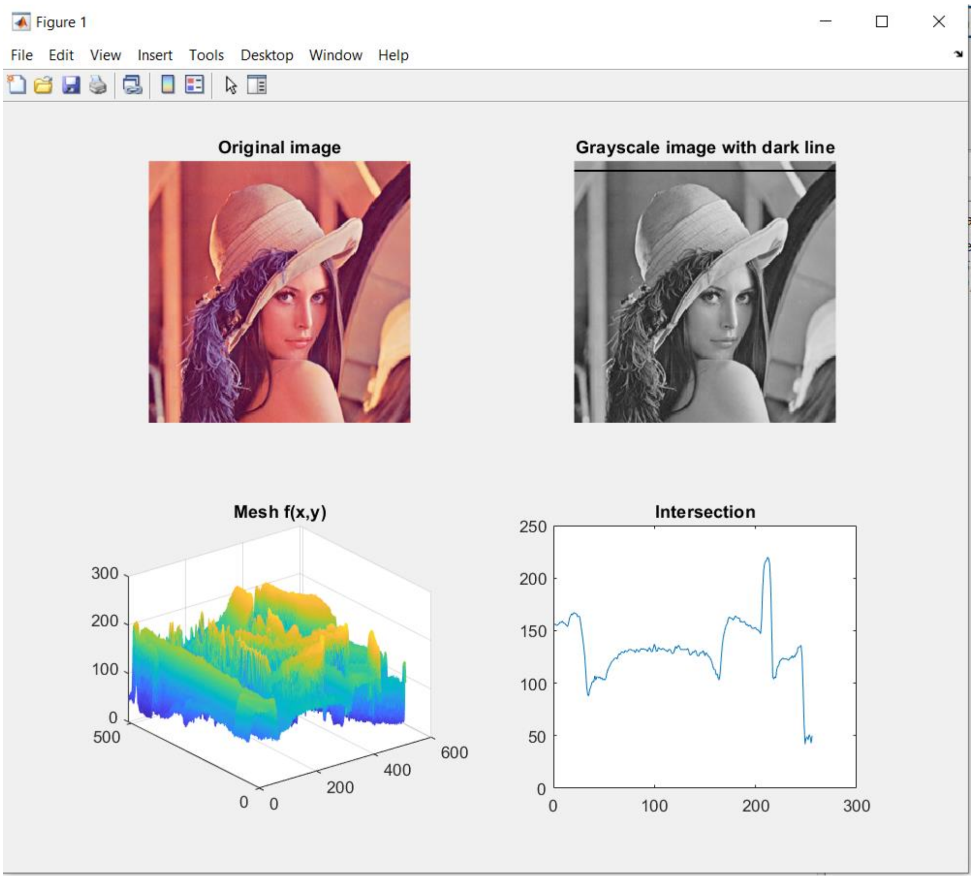
d) Plot an intersection (“slice”) of the grayscale version of the image, i.e. values of f(x,y) for when either x or y is constant. Use plot function, e.g. plot(A(10, :)) to plot row 10 and all columns. Display the result.

```
subplot(2,2,4);
plot(B(10,:));
title('Intersection');
```

f) *Mark the line of intersection by replacing row 10 of the grayscale image by zeros (dark line) and modify subplot with the grayscale image in point (e).

```
subplot(2,2,2);
B(10,:)= zeros();
imshow(B);
title('Grayscale image with dark line');
```

Result of the code:

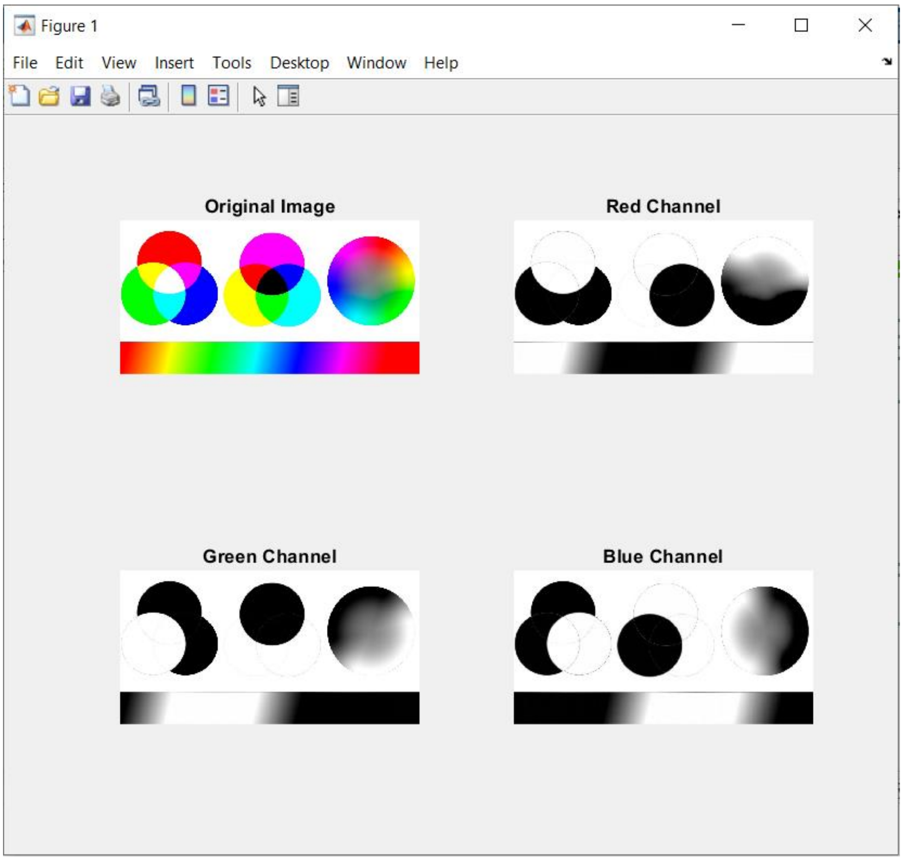


I-3 Colour spaces

a) Create a new section (%%). Load image colours.jpg and display each colour component of R, G, B in a subplot. For example the fist channel can be assigned by: R = A(:, :, 1); Add titles for all subplots.

```
image = imread('Colours.jpg');

R = image(:, :, 1);
G = image(:, :, 2);
B = image(:, :, 3);

figure;

subplot(2, 2, 1);
imshow(image);
title('Original Image');

subplot(2, 2, 2);
imshow(R);
title('Red Channel');

subplot(2, 2, 3);
imshow(G);
title('Green Channel');

subplot(2, 2, 4);
imshow(B);
title('Blue Channel');
```

Result of the code:



Original Image

Red Channel

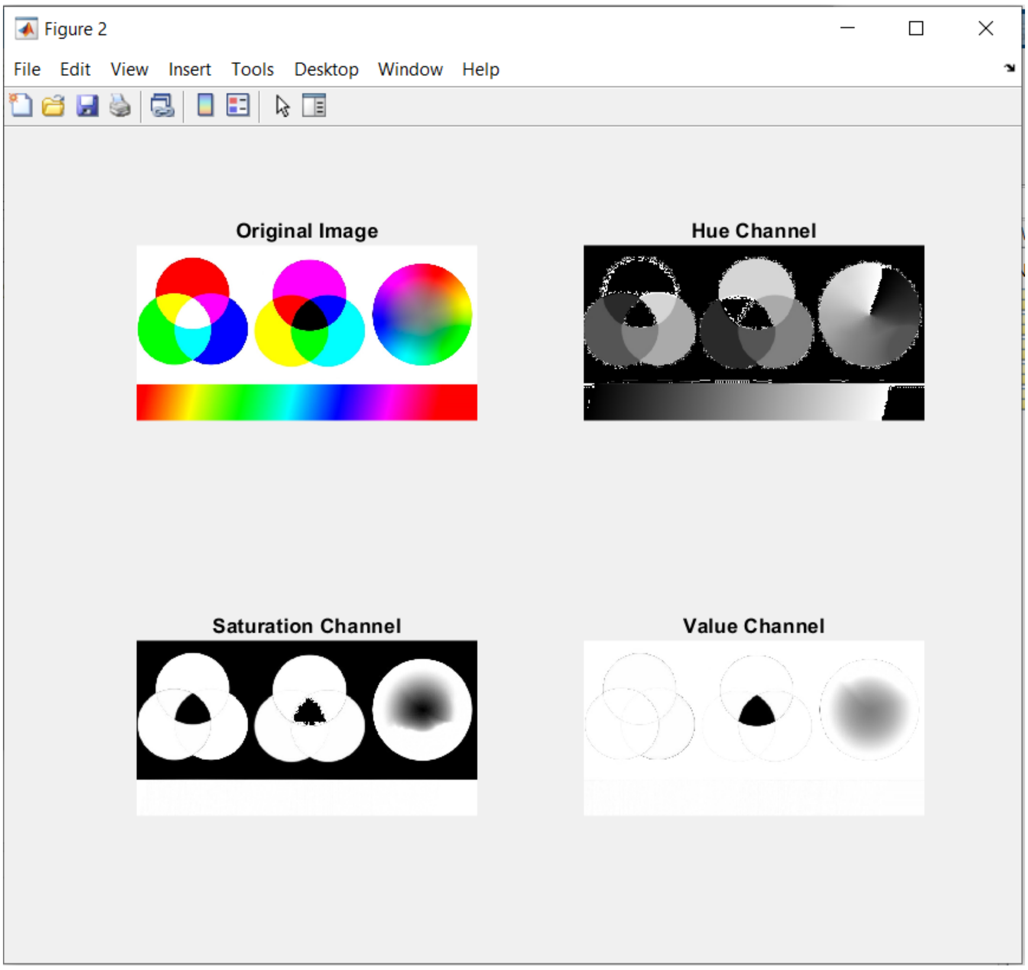
Green Channel

Blue Channel

b) Convert the image to HSV (rgb2hsv) and display the original image and three channels:

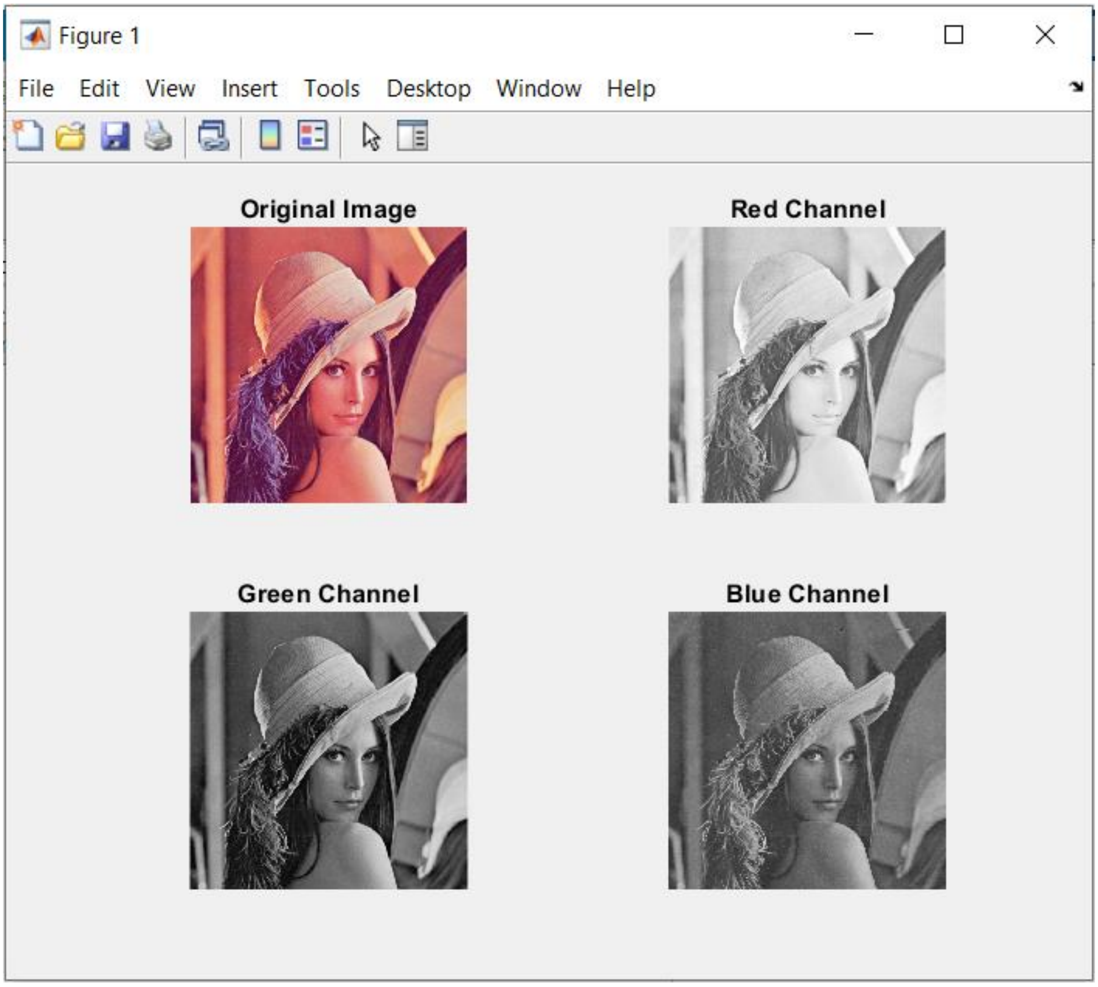
```
hsvImage = rgb2hsv(image);

figure;

subplot(2, 2, 1);
imshow(image);
title('Original Image');

subplot(2, 2, 2);
imshow(hsvImage(:, :, 1));
title('Hue Channel');

subplot(2, 2, 3);
imshow(hsvImage(:, :, 2));
title('Saturation Channel');

subplot(2, 2, 4);
imshow(hsvImage(:, :, 3));
title('Value Channel');
```

Result of the code:



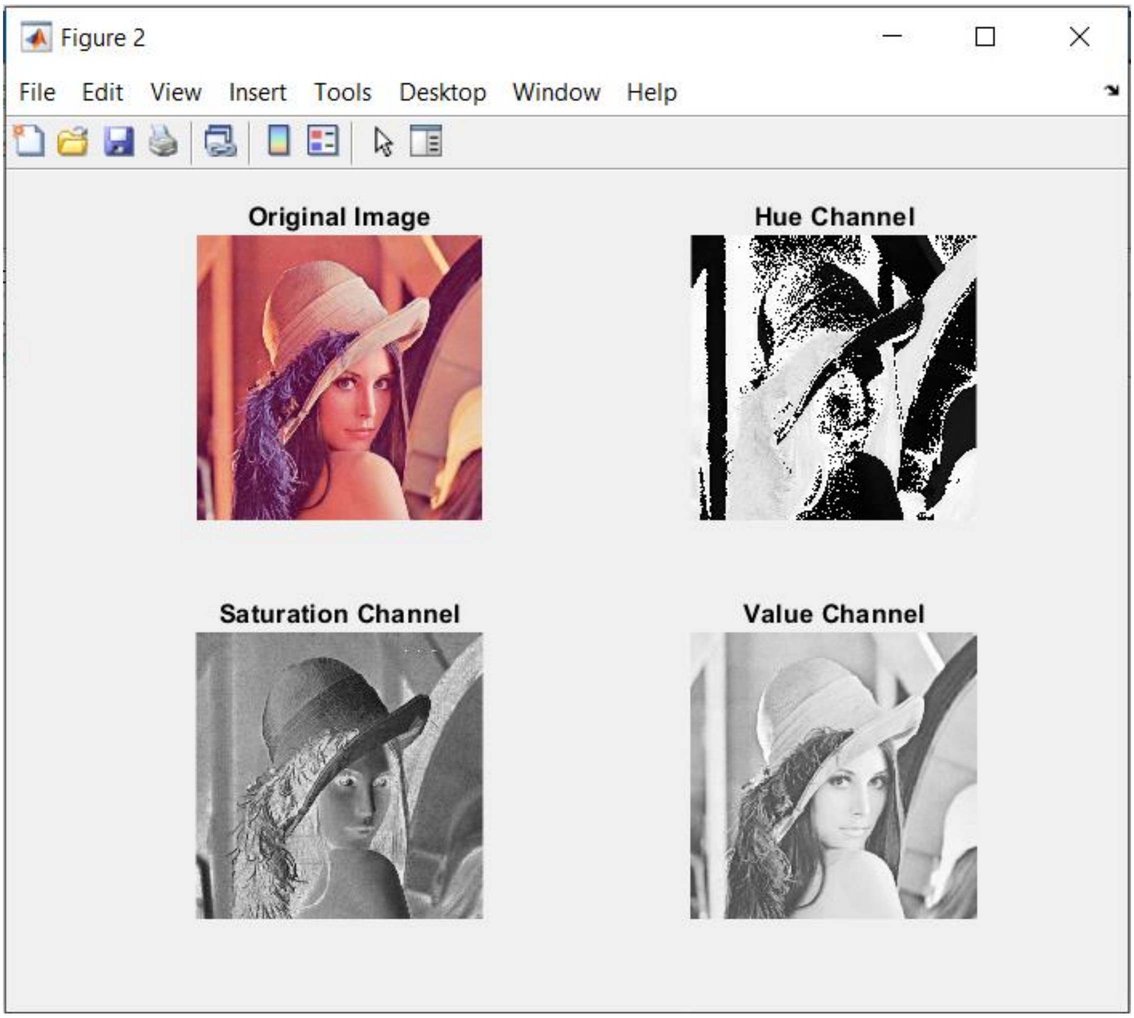
```
c) Perform (a) and (b) for lena.bmp and observe the results.

image = imread('lena.bmp');

R = image(:, :, 1);
G = image(:, :, 2);
B = image(:, :, 3);

figure;

subplot(2, 2, 1);
imshow(image);
title('Original Image');

subplot(2, 2, 2);
imshow(R);
title('Red Channel');

subplot(2, 2, 3);
imshow(G);
title('Green Channel');

subplot(2, 2, 4);
imshow(B);
title('Blue Channel');
```

Result of the code:



```
%----------------------------

hsvImage = rgb2hsv(image);

figure;

subplot(2, 2, 1);
imshow(image);
title('Original Image');

subplot(2, 2, 2);
imshow(hsvImage(:, :, 1));
title('Hue Channel');

subplot(2, 2, 3);
imshow(hsvImage(:, :, 2));
title('Saturation Channel');

subplot(2, 2, 4);
imshow(hsvImage(:, :, 3));
title('Value Channel');
```

Result of the code:



*d) * (optional task) Do the same for colour spaces YCbCr and CIE Lab (corresponding conversion functions are: rgb2ycbcr and rgb2lab).*

```
image = imread('Colours.jpg');

ycbcrImage = rgb2ycbcr(image);

figure;

subplot(2, 2, 1);
imshow(image);
title('Original Image');

subplot(2, 2, 2);
imshow(ycbcrImage(:, :, 1));
title('Hue Channel (ycbcr)');

subplot(2, 2, 3);
imshow(ycbcrImage(:, :, 2));
title('Saturation Channel (ycbcr)');

subplot(2, 2, 4);
imshow(ycbcrImage(:, :, 3));
title('Value Channel (ycbcr)');
```

Result of the code:

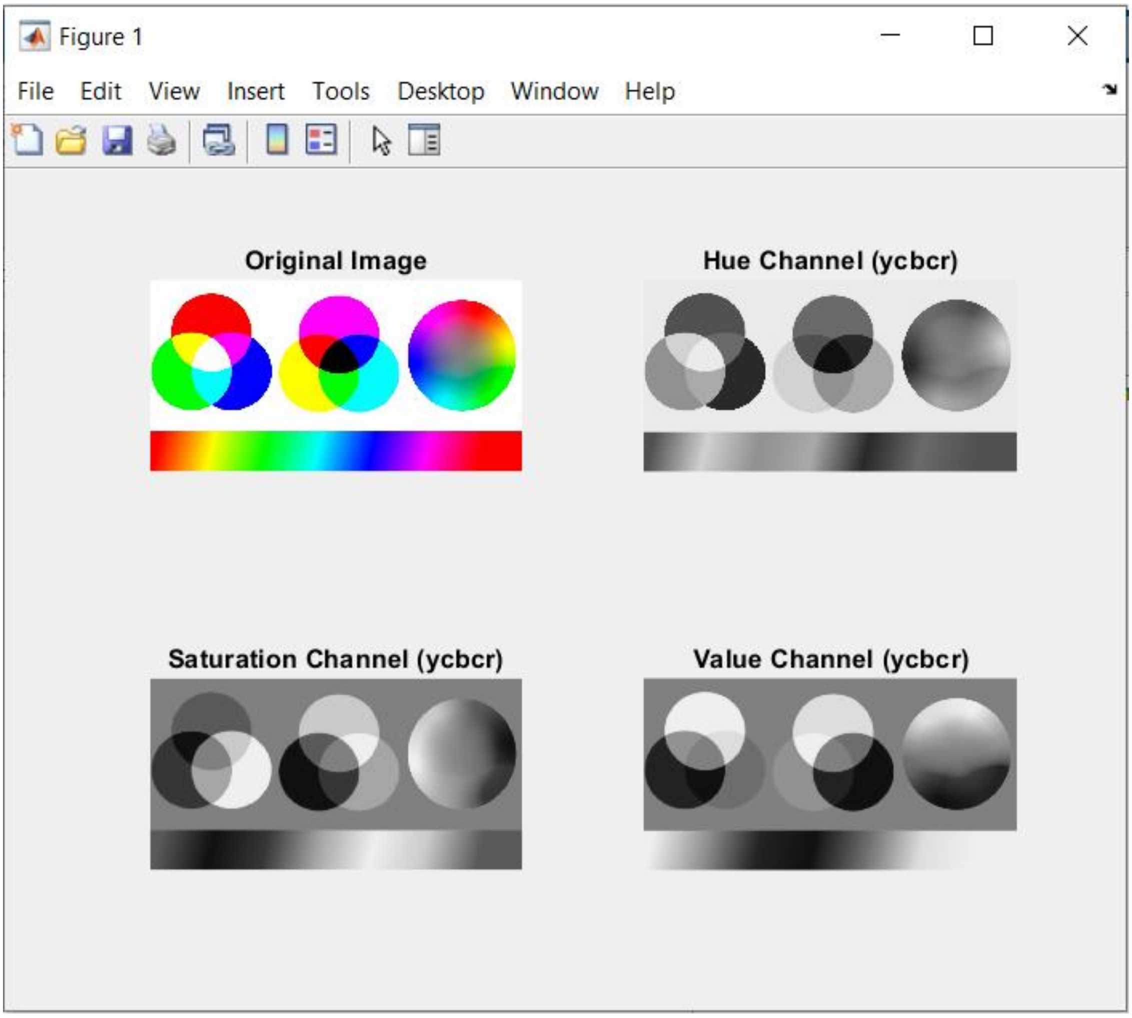


Figure 1
File Edit View Insert Tools Desktop Window Help
Original Image
Hue Channel (ycbcr)
Saturation Channel (ycbcr)
Value Channel (ycbcr)

```
% ---------------------------------

labImage = rgb2lab(image);

figure;

subplot(2, 2, 1);
imshow(image);
title('Original Image');

subplot(2, 2, 2);
imshow(labImage(:, :, 1));
title('Hue Channel (lab)');

subplot(2, 2, 3);
imshow(labImage(:, :, 2));
title('Saturation Channel (lab)');

subplot(2, 2, 4);
imshow(labImage(:, :, 3));
title('Value Channel (lab)');
```

Result of the code:

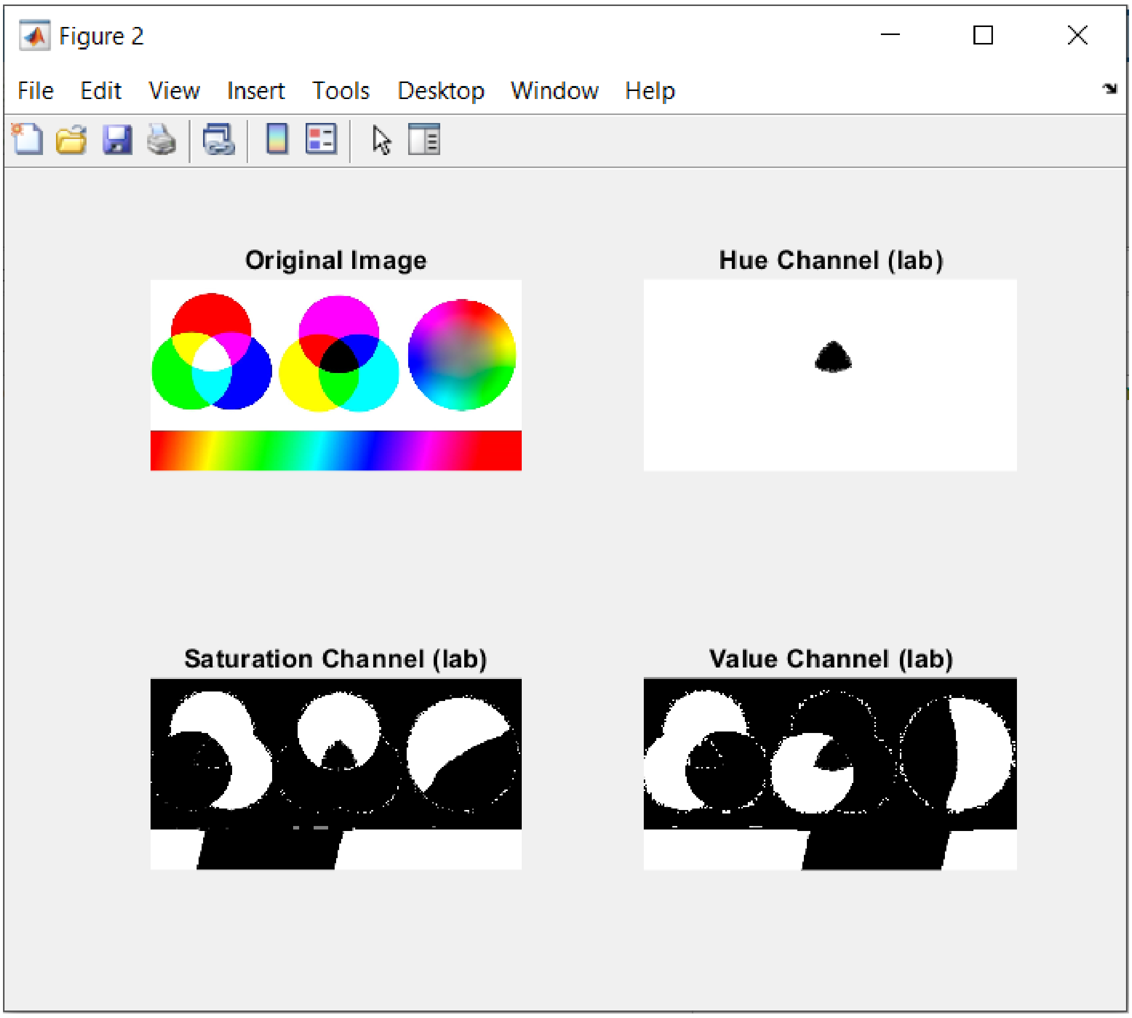


Figure 2
File Edit View Insert Tools Desktop Window Help
Original Image
Hue Channel (lab)
Saturation Channel (lab)
Value Channel (lab)