# Lab #7

## Clustering (24)

In this lab, we will load an image and perform K-means clustering in both RGB and HSV colour spaces.

1. Load the image *tulips.jpg* using the MATLAB function `imread`. **(1)**

2. View the image using `imshow` (use `figure` to open a new figure window).` **(1)**

3. Convert the image from integer data type to double precision float data type and rescale to 0-1. **(1)**

   ```
   img = double(img)/255;
   ```

4. Reshape the image data to be a list of all pixels where each row is a single pixel with 3 columns for the colour channels. **(1)**

   ```
   pix = reshape(img, [size(img,1) * size(img,2),3]);
   ```

5. Perform K-means clustering on the pixel data using `kmeans` and a set number of clusters. **(1)**

   ```
   [labels,centres] = kmeans(double(pix),numClusters);
   ```

6. Reshape the list of labels back into the shape of the original image to form the clustered image. **(2)**

   ```
   imgLabeled = reshape(labels, [??, ??, 1]);
   ```

   Note: You determine the ?? values. Hint: check the size the labels vs. the size of the original image with numel() function;

7. Show the clustered image, using `colormap` to assign the cluster centre colours to each label. **(1)**

   ```
   imshow(imgLabeled);
   colormap(centres);
   ```

8. Repeat steps 5-7 for 2 additional numcluster values and present all 3 k-means anlaysis in a 1x3 plot. **(4)**

9. Now try the entire process but using HSV colour space data instead of RGB. Repeat step 4 but first convert the image to HSV using `rgb2hsv`. **(4)**

10. Before simply applying K-means, remap the 3D locations in HSV space to match the cylindrical coordinate system we've seen used to describe HSV – hue is the angle around the cylinder, saturation is the distance from the centre, and value is the vertical location. **(1)**

```
x = pix(:,2).*cos(2*pi*pix(:,1));
y = pix(:,2).*sin(2*pi*pix(:,1));
z = pix(:,3);
pixMapped = [x y z];
```

11. Now apply K-means clustering to the mapped HSV pixel data. **(3)**

12. The resulting centres need to be unmapped back to HSV space and then converted back to RGB space. **(2)**

```
ctrsHue = atan2(centres(:,2),centres(:,1));
ctrsSat = sqrt(centres(:,1).^2 + centres(:,2).^2);
ctrsVal = centres(:,3);
centresRGB = hsv2rgb([ctrsHue ctrsSat ctrsVal]);
```

13. Repeat steps 6 and 7 to view the resulting clustered image. **(2)**