Lab 8

Kmeans clustering and Hough transform

Objectives:

- Apply the kmeans clustering algorithm to solve segmentation problems
- Use Ostu thresholding for segmentation .
- Extract lines using Hough transform.

Experiment 1: Array Indexing

We experiment how to index the array using an array of indexes.

Experiment 2: apply Kmeans clustering in color segmentation

In this exercise we experiment built in function Kmeans that classify the image into K regions.

1. Read the image "***balls.jpg***".
2. To start clustering, you must first decide if you will cluster based on the [R/G/B channel or all of them this depends on the image and the problem you are trying to solve], also the same thing applies to HSV channels.
3. The ***"cluster.KMeans"*** function takes number of clusters argument (n_clusters) and returns a kmeans_cluster object.
4. To get the kmeans call kmeans_cluster.fit() with image data argument. It must be in a form of vector not matrix.
5. kmeans_cluster has two important attributes:
    - cluster_centers_ : contains the center value of each centroid (the color for each segment).
    - labels_ : contains the label for each pixel.

6. Use the same method described above to get the new value of each pixel according to each label.
7. Reshape the matrix to be the same image as the original image.
8. Convert the image to uint8 type.
9. Show the images
10. Conduct the results to the three images '***balls.jpg'***, ***'amazing.jpg'***, '***colors.jpg'***, choose best K and choose best channel/channels to cluster on them.

Experiment 3: Line Extraction using Hough Transform

Read the comments in the notebook, and implement each line start with [TO-DO] then answer any commented question.

Experiment 4(bonus): Apply clustering using otsu threshold:

In this exercise we do clustering and segmentation in gray level using thresholds and labels, then use these labels to cluster the colored image into different clusters, as follows:

1. Choose a channel or gray-scale image to separate on.
2. Get threshold of grey level image to separate background from objects using the function ***"threshold_otsu"***.
3. Replace all values less than threshold with black and values more than threshold with white.
4. Show original image and the binary image

5. Conduct the results to the three images '***balls.jpg'***, ***'amazing.jpg'***, '***colors.jpg'***, choose best channel/channels to cluster on them.
6. Compare the results to Exercise one.