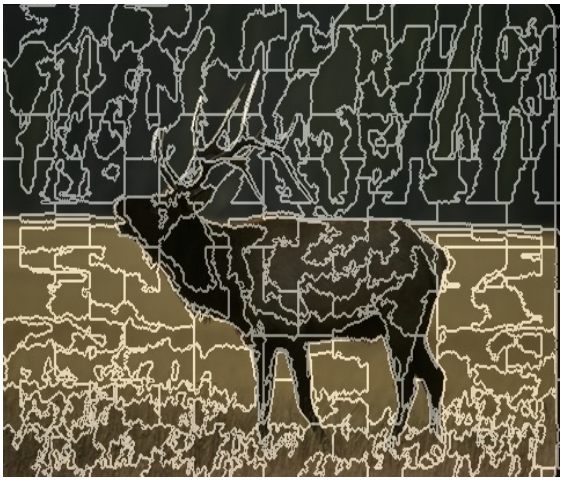
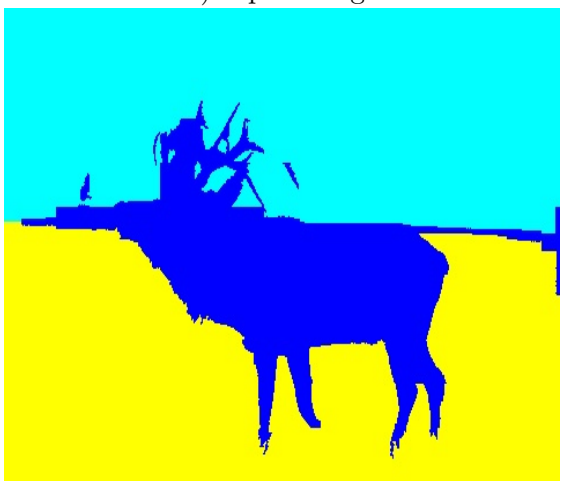
Assignment 3

Due on December 26, 2023 (23:59:59)

a) Input Image



b) Superpixels



c) Image Labels

d) Image Segments

Figure 1: K-Means Image Segmentation using Superpixels

K-Means Clustering for Image Segmentation

Background

Clustering is a process that groups data with respect to data similarity so that similar data take part same cluster. In image domain clustering is used for various types of problem e.g. *image quantization*, *image segmentation*. A good clustering algorithm must group data to homogeneous subsets as possible. Similarity is most critical step

in a clustering algorithm that determine how the clustering algorithm groups data. K-means clustering is one of the most popular clustering algorithm that groups data to k dissimilar clusters. It is an unsupervised learning algorithm for clustering problem and the main idea is to define k centroids one of each cluster. These centroids is randomly assigned to data space for first iteration. In the next steps, for each data point, distance to these centroids is calculated and data points are assigned to nearest centroids as cluster elements. Then for each cluster, new k centroids are calculated from k clusters. This steps go on until clusters centroids unchanged.

Overview

In this assignment, you will use K-means clustering algorithm for image segmentation by using pixel level and superpixel representation of an input image. For this purpose you must carry out the following steps:

1. Extract feature : Extract features for input image by using the definitions below.
2. Perform K-means clustering : Segment image by using extracted features with optimal k parameter.

You'll extract features that listed below and you'll perform segmentation method by using each of them then you'll make comments about results.

1. **Features**

   - **Pixel-Level Features**
     You'll extract two different features for every pixel in the image; RGB color feature and spatial location feature.

     (a) RGB color feature: You should concatenate R, G, B color channel values for each pixel for representation. In other words each pixel will be represented with [R G B].

     (b) RGB color and spatial location feature: Every pixel should be represented with RGB color values and location information which is the coordinate of the pixel. Each pixel will be represented with [R G B x y] feature vector.
     Hint: Be careful, color and location values are from different range of numbers. You should normalize your features before concatenation. In other words you should map your RGB and location features to the same value interval.

   - **Superpixel-Level Features**
     For this step you have to extract superpixels by using SLIC Superpixel work, for which, alternatively you can use "slic()" function in from scikit-image library. You will define a feature vector to represent each superpixel.

(a) Mean of RGB color values: A superpixel is represented with mean color value of pixels that are included by the superpixel.

(b) RGB color histogram: A superpixel is represented by RGB color histogram which is extracted by using all pixels contained by that superpixel.

(c) Mean of Gabor filter responses: At this step you'll create a filterbank by calculating Gabor filters at different scales and orientations. Then you'll filter input image with each Gabor filter. You'll use the response map to represent superpixels. A superpixel is represented by calculating mean of Gabor filter response values of pixels that are included by the superpixel. Alternatively you can use Gabor from scikit-image.

2. **K-Means Clustering**
   You have to implement your own K-Means clustering algorithm for this step. Your K-means function has to take two parameters; data matrix(feature matrix/vector) and k parameter. You'll use your feature matrix as input data and determine k parameter which is the count of cluster you want to generate.

   You must perform clustering for each feature; RGB color at pixel-level, RGB color and location feature at pixel-level, mean RGB feature at superpixel level, RGB color histogram at superpixel level, mean Gabor response at superpixel level.

Details

Your program will take an color and cluster number as input and produce segmentation result like in Figure 1-c,1-d.

- You must take results on the least 5 images. You can find lots of image in Berkeley Segmentation Dataset[1].
- You must play with cluster number according to the image content to obtain good results.
- You must use superpixels with different region size and make observation that how the superpixels and region size of superpixels influence clustering results or not.
- You must show your results like in Figure 1-c,1-d. You can use Matplotlib library for this purpose.
- You must comment about your results why they are satisfactory or not.
- You must comment about pixel level and superpixel level results.
- You must comment about features that are used (comment about features' efficiency and representative power).
- **You have to implement your own K-means clustering method.**

What to Hand In

You are required to submit all your report along with PDF format. For that purpose, prepare a folder containing:

- README.txt *(text file containing details about your assignment and this file also must include a Google Drive link (**without forcing to login**) including all your images (Input Image, Superpixels, Image Labels, Image Segments))*
- code/ *(directory containing all your code (only ".py" files))*
- report.pdf *(PDF report)*

Archive this folder as **studentid_pset3.zip** and send to **submit system**.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] `https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds`