ELEC 474

Lab 2 Image Segmentation

K-Means Clustering

Kevin Huang and Michael Greenspan

Contents

1. K-Means Clustering and Color Quantization

For Lab 2 you will be implementing a *k-means clustering* algorithm. You will perform *color quantization* on an image using k-means with random center(s) initialization. The goal of color quantization is to reduce the number of colors in an image. K-means clustering aims to partition n image pixels to their corresponding k clusters based on the nearest means. For this lab, you will initialize k random cluster center(s), $\{ c_i \}, i = 1..k$.

The steps to perform k-means clustering are as follows:

K-Means Initialization:

1. First **randomly** select k points within the image bounds. You should create a track bar to select and experiment with different values of k. Your randomly selected initial cluster centers, $\{ c_i \}, i = 1..k$, should have associated color channel intensities, as they will be used to calculate your **means**.

2. Partition your image into ***Voronoi Regions*** based on the color values of each pixel. The Voronoi Region around each cluster center c_i includes all pixels that are closest to that cluster center, according to some **distance metric**. In this case, as we are only interested in the color values of the pixels, you should use a **distance metric** based solely on the color channels to determine which pixels belong to which cluster.

3. Find the mean color value m_i for each Voronoi Region.

   a) If for each region, the mean values are very close to the cluster centers (i.e. $m_i \cong c_i$ *for all* $i = 1..k$), then the algorithm has converged. Recolor all pixels in each region to their mean color values, and **STOP**.

   b) Else, if any of the mean values differ significantly from their cluster centers (i.e. $m_i \ncong c_i$ *for any* $1..k$), then update the cluster centers to equal the mean values (i.e. set $c_i = m_i$) and repeat Steps 2 and 3.

Notes:

A. For Step 2, you will need to implement a similarity metric to determine how close two pixels are. This can be as simple as the sum of the squared difference of the three color channel values.

B. For Step 3, use the numpy ***mean()*** method to calculate the mean of the region.

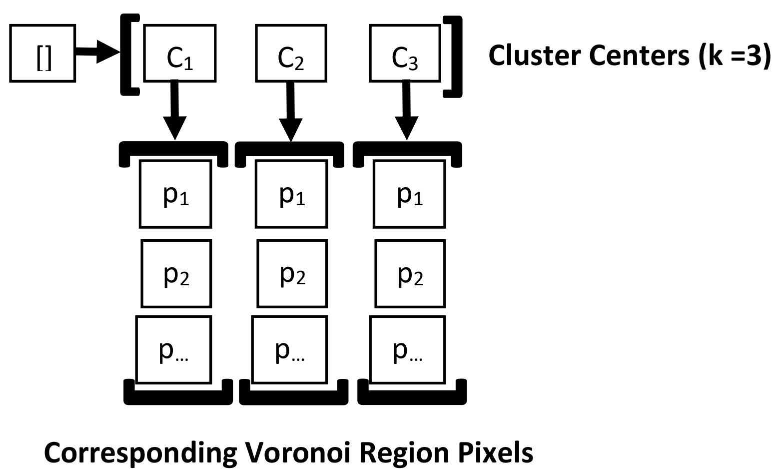
C. For Step 3 a), you can reuse the similarity metric from Note A above, and compare its value to a threshold to determine how close the mean is so the cluster center.

D. You should have three main methods implemented in this order:
- Random Center Initialization based on the number of clusters $\boldsymbol{k}$.
- Voronoi Region Segmentation based on the set of cluster centers.
- KMeans which calls Voronoi Region Segmentation.

It would be helpful to implement and test these functions separately as each depends on the previous.

E. Using OpenCV's ***waitKey()*** would be helpful with key input and listening when debugging and testing functions separately.

F. While there are a number of ways of representing the Voronoi Regions, the use of a 2D point array can be useful to associate region pixels with their corresponding cluster centers:



Here is K-Means Clustering performed on a series of images:

Baboon.jpg with K-Means clustering performed at $\boldsymbol{k}$ = 2, 10, and 20 clusters

South African Flag (SAFlag.jpg) with K-Means clustering performed at $\boldsymbol{k}$ = 3, 8, and 22 clusters

Keep in mind that these images have **randomly initialized centers**, so these processed images will differ from your images based on your $\boldsymbol{k}$ and selected $\boldsymbol{c}$.

2. Submission

The submission for Lab 2 should contain *.ipynb file that includes your code that performs K-Means Clustering for Color Quantization on both the “baboon.jpg” and “SAFlag.jpg” images.

Your code will be run in Jupyter Lab to test for functionality.

When submitting your lab, make sure that you follow the formatting instructions that are listed on the onQ site under ***‘Resources->lab submission instructions’.***

The marking rubric is as follows:

Item	**mark**
1. Voronoi region calculation and partition correct	1
2. Similarity metric correct	½
3. K-means algorithm implemented and executes correctly	2
4. Submission format correct	½
Total:	**4**