SLIC Project Report

Simple Linear Iterative Clustering (SLIC), Computer Vision

Prerna Agarwal

Winter 2019

ABSTRACT

A lot of computer vision applications involve image segmentation. Image segmentation refers to partitioning an image into multiple segments based on similar attributes to simplify image analysis. One such type of segmentation is color based segmentation or also known as superpixels. In order to understand this type of segmentation, I re-implemented Simple Linear Iterative Algorithm (SLIC) based on “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods” paper. I chose this algorithm for its ease of simplicity.

INTRODUCTION

Superpixel algorithm groups pixels into various non-overlapping superpixels i.e., perceptually meaningful regions while respecting the potential object contours. Thus, it can replace the original rigid pixel grid structure. This method of segmentation is important for image analyses as it is computationally efficient i.e., instead of computing over hundreds of thousands of pixels we can just compute over few hundreds of pixels and it is perceptually meaningful i.e., it provides a convenient primitive form to analyze the image and reduces redundancy . Due to the simplistic nature of superpixels, they are applied to various computer vision tasks likes multiclass object segmentation, depth estimation, object localization etc.

METHOD

One of the methods to implement superpixel segmentation is Simple Linear Iterative Clustering (SLIC). It is an adaptation of K means clustering and requires very less computational power. It performs a local clustering of pixels in 5-D space defined by the L, a, b values of the CIELAB colorspace and x, y coordinates of the pixels. It generates compact and nearly uniform superpixels by clustering the pixels based on their color and texture similarity and proximity in the image plane.

The only parameter given to the algorithm is K which is the desired number of superpixels. To produced approximately equally sized superpixels, each superpixel will have approximately N/K pixels with center Ck = [lk, ak, bk, xk, yk] where k = [1, K] and grid interval $S = \sqrt{(N/K)}$. The centers are moved to seed locations corresponding to the lowest gradient position in a 3X3 neighborhood in order to avoid centering a superpixel

on an edge and to reduce the probability of selecting a noisy superpixel center.

In order to increase the speed of SLIC over regular k means algorithm, the pixels assigned to a cluster lie within 2S x 2S area around the superpixel center in the xy plane. This is done to limit the size of the search region, reduce the number of distance calculations and avoid comparing each, pixel to every cluster center to find it's best match cluster center.

Since the cluster centers are composed of CEILAB color space (l, a, b) and pixel's position (x, y), we can't directly take the euclidean distance between the center and a pixel. This will result in inconsistencies for different superpixel sizes. For larger superpixels, if spatial pixel distance exceed the perceptual color distance limit, then they begin to outweigh pixel color similarities and the opposite is true for smaller pixels. Thus, we need to normalize the color proximity and space proximity to get a combined distance of a pixel from the center.

- $d_{lab} = \sqrt{((l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - bi)_2)}$
- $d_{xy} = \sqrt{((x_k - x_i)^2 + (y_k - y_i)^2)}$
- $D_{combined} = \sqrt{(d_{lab}/m)^2 + (d_{xy}/S)^2}$

Where m is a constant between 1 to 20.

Once all the pixels are associated with the nearest cluster center, a new center is computed by averaging over the cluster members and a residual error is also computed between the previous center and the new center. We keep iterating through this algorithm until the max number of iterations is reached i.e., 15.

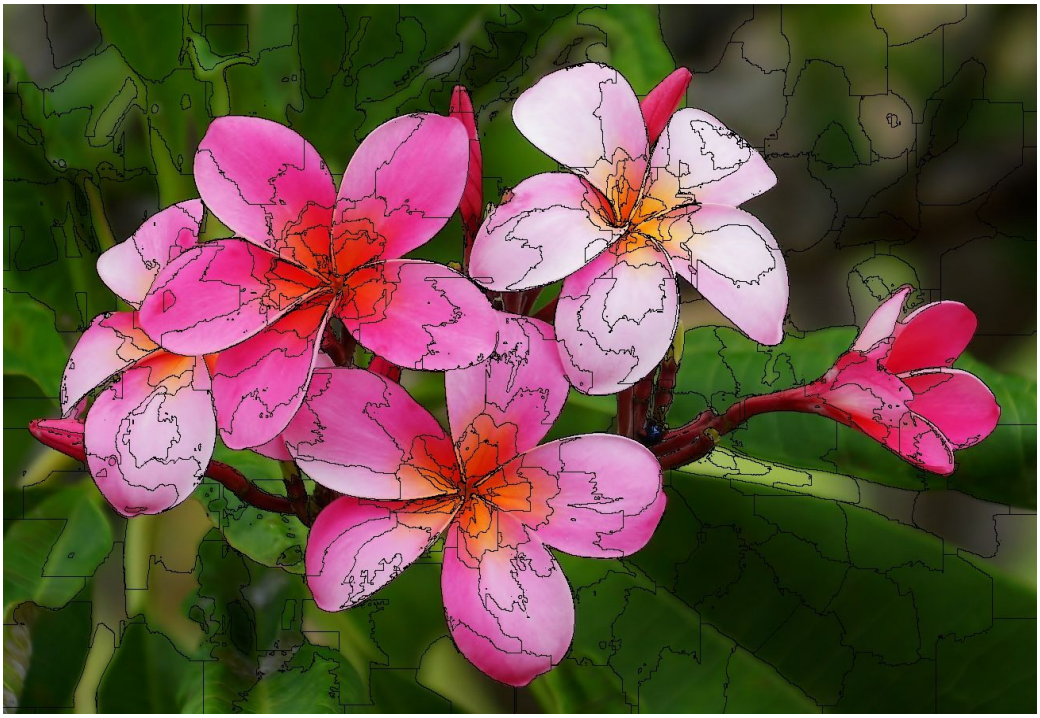
RESULT

I downloaded two sample images from the internet and ran SLIC algorithm for k = 64, 256 and 1024 superpixels (or number of clusters). The results are as expected. We can see when K is less, superpixels have random shapes and are distinctly formed on the basis of color and texture similarities. However, when K increases, superpixels start depicting grid like structures and the color distinction between nearby superpixels becomes less noticeable by a human eye. Moreover, if the image and K are not known in advance, SLIC may also lead to over segmentation. The results are as follows:

Flower:

K = 64:

K = 256:

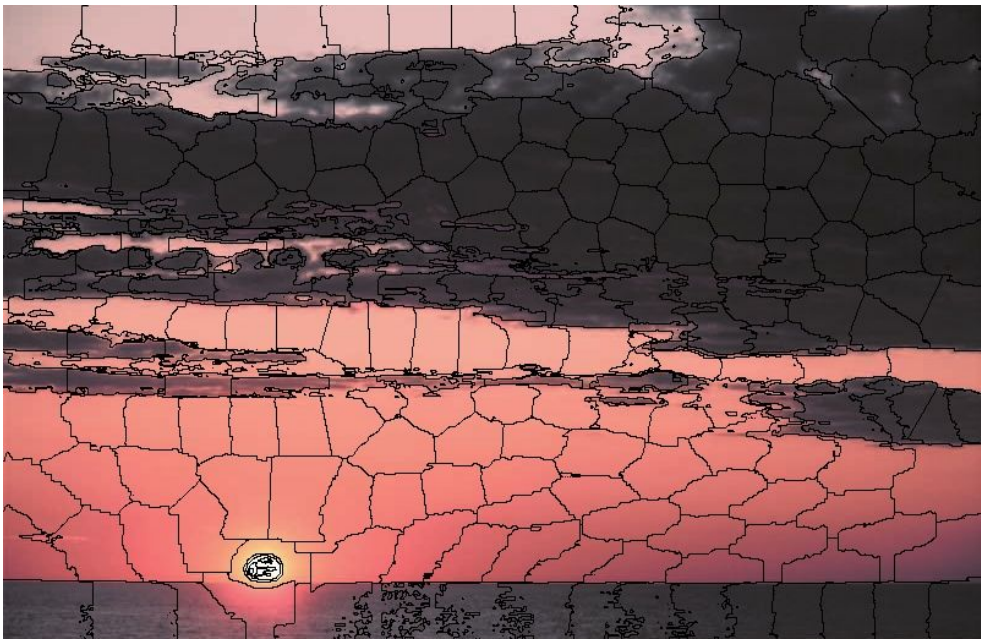


K = 1024:

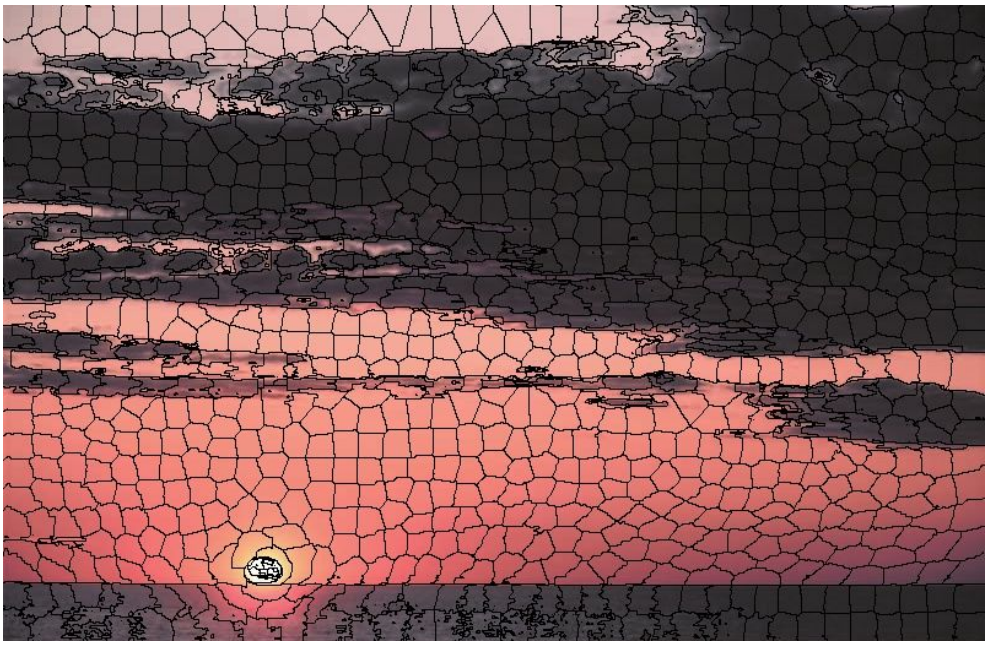
Sunset:

K = 64

K = 256



K = 1024



REFERENCES

1. Achanta, Radhakrishna, et al. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods.” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, 2012, pp. c3–c3., doi:10.1109/tpami.2012.202.
2. Achanta et al. “PSMM/SLIC-Superpixels.” GitHub, PAMI'12, Vol. 34, Num. 11, Pp. 2274-2282, 13 May 2013, github.com/PSMM/SLIC-Superpixels.
3. Körting, Thales Sehn, director. YouTube. YouTube, YouTube, 1 Sept. 2018, www.youtube.com/watch?v=-hmUbB-Y8R0
4. Mallick, Satya. “Home.” Learn OpenCV, 5 Nov. 2018, www.learnopencv.com/image- segmentation/
5. Rambhia, Jay. “SLIC Based Superpixel Segmentation.” 25 Aug. 2013, jayrambhia.com/blog/superpixels-slic
6. Rosebrock , Adrian. “Segmentation: A SLIC Superpixel Tutorial Using Python.” PyImageSearch, 2 Aug. 2018. www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/